Modelling and Model Based Control Design

For Rotorcraft Unmanned Aerial Vehicle

by

Rejina Choi Ling Wei

Department of Mechanical
Canterbury University

Thesis submitted in partial fulfilment of
the requirements for the degree of Doctor
of Philosophy in the Department of
Mechanical in the Graduate School of
Canterbury University

2014

Abstract

Designing high performance control of rotorcraft unmanned aerial vehicle (UAV) requires a mathematical model that describes the dynamics of the vehicle. The model is derived from first principle modelling, such as rigid-body dynamics, actuator dynamics and etc. It is found that simplified decoupled model of RUAV has slightly better data fitting compared with the complex model for helicopter attitude dynamics in hover or near hover flight condition. In addition, the simplified modelling approach has made the analysis of system dynamics easy. System identification method is applied to identify the unknown intrinsic parameters in the nominal model, where manual piloted flight experiment is carried out and input-output data about a nominal operating region is recorded for parameters identification process. Integral-based parameter identification algorithm is then used to identify model parameters that give the best matching between the simulation and measured output response. The results obtained show that the dominant dynamics is captured. The advantages of using integral-based method include the fast computation time, insensitive to initial parameter value and fast convergence rate in comparison with other contemporary system identification methods such as prediction error method (PEM), maximum likelihood method, equation error method and output error method. Besides, the integral-based parameter identification method can be readily extended to tackle slow time-varying model parameters and fast varying disturbances. The model prediction is found to be improved significantly when the iterative integral-based parameter identification is employed and thus further validates the minimal modelling approach.

From the literature review, many control schemes have been designed and validated in simulation. However, few of them has really been implemented in real flight as well as under windy and severe conditions, where unpredictable large system parameters variations and unexpected disturbances are present. Therefore, the emphasis on this part will be on the control design that would have satisfactory reference sequence tracking or regulation capability in the presence of unmodelled dynamics and external disturbances. Generalised Predictive Controller (GPC) is particularly considered as the helicopter attitude dynamics control due to its insensitivity with respect to model mismatch and its capability to address the control problem of nominal model with dead-time. The robustness analysis shows that the robustness of the basic GPC is significantly improved using the Smith Predictor (SP) in place of optimal predictor in basic GPC. The effectiveness of the proposed robust GPC was well proven with the control of helicopter heading on the test rig in terms of the reference sequence tracking performance and the input disturbance rejection capability. The second motivation is the investigation of adaptive GPC from the perspective of performance improvements for the robust GPC. The promising experimental results prove the feasibility of the adaptive GPC controller, and especially evident when the underlying robust GPC is tuned with low robustness and legitimates the use of simplified model. Another approach of robust model predictive control is considered where disturbance is identified in real-time using an iterative integral-based method.

Dedication

This thesis is dedicated to my mother.

Acknowledgements

First, I would like to thank Professor Dr. XiaoQi Chen, who offered me the opportunity to come to University of Canterbury and gave me the assistance I needed to accomplish this. I would like to express my utmost gratitude to my co-supervisor Dr. Christopher E. Hann for his guiding, help and support in my research.

I would also like to thank my research fellows in Mechatronics Research Laboratory, for their advice and their help during the research. I would like to thank Syariful Syafiq Shamsudin for leading me into the field of unmanned aerial vehicle, and introducing me to LabVIEW programming as well as National Instrument development kits. I would like to thank Mervin Chandrapal in giving me useful ideas and guidance in constructing test rig for helicopter control development. I would like to thank the technicians in Mechanical Engineering Department, University of Canterbury, in particular, Julian Murphy and Gerry Cotton for their invaluable assistance and support during my time of research here in University of Canterbury.

My special thanks to my family who provide me financial support, understanding and encouragement throughout my studies. I would like to express my gratitude to my mother who always gives me moral support when I faced with difficulties in my research and help me to get over my downtime and difficult moments.

Publication List

The following is a list of my publications that have been published during the period of my doctoral study:

Conference Papers

- Rejina L.W Choi, Christopher E. Hann and Xiaoqi Chen. 'Minimal Models to Capture the Dynamics of a Rotary Unmanned Aerial Vehicle'. *19th International Conference on Mechatronics and Machine Vision In Practice (M2VIP),* Auckland University of Technology, Auckland, New Zealand, 28-30 November 2012, pp. 91-96.

Journal Papers

- Rejina L.W Choi, Christopher E. Hann and Xiaoqi Chen 'Minimal Models to Capture the Dynamics of a Rotary Unmanned Aerial Vehicle'. *Journal of Intelligent & Robotic Systems:* Volume 75, Issue 3 (2014), Page 569-593. DOI: 10.1007/s10846-013-9993-5
- Rejina L.W Choi, Christopher E. Hann and Xiaoqi Chen 'Iterative parameter estimation and model prediction of a rotary unmanned aerial vehicle'. *Int. J. of Intelligent Systems Technologies and Applications*, 2014 Vol.13, No.1/2, pp.81 - 102. DOI: 10.1504/IJISTA.2014.059301

Contents

List of Tables

List of Figures

Nomenclature

AFCS	Automatic Flight Control System
ARX	Auto Regressive with eXtra inputs
ARMAX	Auto Regressive Moving Average with eXtra inputs
ANN	Artificial Neural Network
CARIMA	Controlled Auto-Regressive Integrated Moving Average
CIFER	Comprehensive Identification From Frequency Response
DCM	Direction Cosine Matrix
DOF	Degree Of Freedom
DMC	Dynamic Matrix Control
DTC	Dead-time Compensator
EKF	Extended Kalman Filter
ESC	Electronic Speed Controller
FFT	Fast Fourier Transform
FIR	Finite Impulse Response
GNC	Guidance Navigation Control
GPC	Generalised Predictive Control
GPS	Global Positioning System
IMU	Inertial Measurement Unit
LQR	Linear Quadratic Regulator
MAV	Micro Air Vehicle
MPC	Model Predictive Control

MRAC	Model Reference Adaptive Control
NLR	Non-Linear Regression
PID	Proportional Integrative Derivative controller
PLR	Pseudo Linear Regression
PWM	Pulse Width Modulation
RC	Remote Control
RHC	Receding Horizon Control
RLS	Recursive Least Square
RPEM	Recursive Prediction Error Method
RPM	Rotation per Minute
SISO	Single Input Single Output
SP	Smith Predictor
TTL	Transistor-transistor Logic
UAS	Unmanned Aircraft System
UAV	Unmanned Aerial Vehicle
UARTs	Universal Asynchronous Receiver/Transmitte

1. INTRODUCTION

In the past unmanned aircraft systems (UAS) were mostly used in military for defense-related applications. The future military scenario has also become the main investment drive for UAS research. However, the trend of using UAS in civilian mission is slowly emerging, starting with needs from government organization for surveillance and inspection systems, such as coast guards, border patrol organizations, emergency services, and police. The studies on UAS market in civilian applications were done by (Dickerson, 2007; London, 2009), which predict that the UAS market will expand significantly in the next decade.

The rotorcraft UAS (RUAS) especially suit for civilian applications compared with fixed-wing UAS. Civilian applications such as infrastructure inspection, law enforcement, surveillance of coast borders and road traffic, disaster and crisis management, and agriculture and forestry mostly require low altitude hovering flight and vertical take-off and landing capabilities, which are impossible flight manoeuvres for fixed-wing UAS, but special features of RUAS.

Rotorcraft UAS can be categorized into five classes according to their size and payload as shown in Figure 1.

- Category 1: Full-scale unmanned helicopters or optionally piloted autonomous helicopters.
- Category 2: Medium-scaled UAS helicopters that have a significant payload (more than 10kg) with a total weight of more than 30kg.

- Category 3: Small-scale RUAS that are based on RC helicopters with optionally integrated autopilot. They have a payload of several kilograms (2 to 10kg) and a total weight of less than 30kg.
- Category 4: Mini RUAS that are man-portable and can fly in a confined environment as well as outdoors. Their payload is generally less than 2kg and weight from hundreds of grams to a few kilograms.
- Category 5: Micro air vehicles (MAV) have a payload of less than 100g. They normally have unconventional sensing and navigation solutions such as those based on bio-mimetic principles, as standard navigation solutions are too heavy to carry on board.

The research in RUAS has mainly emphasized on three areas, which are guidance, navigation and control (GNC). For a fully autonomous RUAS, the RUAS must process all three systems. Navigation process of RUAS involves monitoring, collecting and processing the information from the surrounding environment and states of vehicle. The observation is compared with RUAS's prior knowledge and is classified into various functions, such as mapping, obstacle and target detection and object recognition. As for guidance system, it is usually associated with trajectory generation, path planning, path generation, and mission planning. Its role is to replace the human pilot in decision making to complete assigned flight mission. Navigation and guidance system are basically higher level of autonomy system in RUAS, which depend on low level automatic flight control to stabilize the vehicle and trajectory tracking. Automatic control system has attracted attention of

many researchers due to its challenging control problem and a platform for testing new control methodology.

Control design for RUAS typically starts with identifying the dynamical model of the vehicle. RUAS are classified as under actuated mechanical systems, due to fewer control inputs than the state variables. In addition, the dynamical model of RUAS is highly nonlinear and has strong coupling between different subsystems. Despite the complexities, having a high-fidelity model enables the simulation of RUAS so that new control methodologies can be tested in computer for performance prior to flight-test.

Figure 1: Categories of unmanned rotorcraft (Kendoul, 2012)

After nearly two decades of active research in the field of UAV, the flight control is mature enough for immediate solutions to RUAS technology needs. Most researchers have now shifted from control-related research to vision-based navigation (Kendoul, 2012).

Nevertheless, there are aspects that require further investigation, such as development of flexible controllers that can be adapted to different platforms in short time and the design of robust controllers that guarantee flight performance under windy and severe weather conditions.

1.1 Research Objectives

As discussed above, rotorcraft unmanned aerial vehicle can be categorized according to their size and payload, naturally their physical geometry parameters will also be varying. However, the same dynamics model can be applied for similar type of aerial vehicle, such as single rotor, quad-rotor, ducted fan and etc. This research will specifically focus on modelling and control of single rotor RC helicopter of category 3.

The objectives of the research are as follows:

- To identify a suitable mathematical model based on RC Trex helicopter. The mathematical model is not only crucial in developing high fidelity controller, but also for simulation purposes to guarantee controller performance before deploying on real flight test. Most often, researchers directly use the mathematical model obtained from first principles for control design, without extensive experimental validation against flight data.
- To design a robust model-based controller that compensates for the disturbance, unmodelled dynamics, and time-varying parameter.
- To test the performance and to validate the practicability of the designed controller with RC helicopter on test rig.

1.2 Scope and Contribution

This research attempts to solve the problem starting with deriving the mathematical model from first principles modelling. However, the mathematical model often possesses many unknown aerodynamics coefficients and geometrical parameters. To identify all the parameters individually would required an in depth knowledge of aerodynamics theory, large amount of wind tunnel testing and parameter tuning. Therefore, system identification approach is taken in this research, where human piloted flight test is carried out in vicinity of a nominal flight envelope and the input output measurements are recorded for model parameters identification as well as for validation. Although the model parameters identified will be in lumped parameter form, however, it is able to reflect the relationship between the different variables and the output. In addition, this research adopts the incremental modelling ideology and provides a flexible framework, where complex dynamics or coupling can be added to the simplified model if it shows improved matching against flight data.

Different methodologies have been proposed by researchers for parameter identification as will be discussed in Section 2.1, and they mostly have the trade-off between computational speed and accuracy. Therefore, the research presented here aims at identifying the unknown parameters at high computational speed with reasonable good fit with flight data. To increase the accuracy of the model matching against flight data, considering the difficulty in obtaining a very accurate model and developing a high performance controller, real time parameters estimation is investigated. With this formulation, the parameter value will be varying at each time step, instead of a static value

that cover the whole data range. The time varying parameters will be able to account for unmodelled nonlinearities, as well as the disturbances effects on the states of vehicle, and thus further extending the flight envelopes. A few algorithms were proposed in the past for real-time parameter estimation (Gutman, 1994; Jakoby and Pandit, 1987; Mutambara and Al-Haik, 1999). However, most of them suffer from high computation load, unknown initial noise statistics, and low parameter convergence rate.

A large variety of flight controllers were designed in the past, ranging from linear flight controller to advanced learning-based flight control systems (Civita et al., 2006; Buskey et al., 2001). However, most of these control systems have been validated in simulation, and very few reported experimental results. In addition, although significant work has been done on developing advanced nonlinear flight controller, most commercial UAV still use autopilots based on linear controllers such as PID, LQR and H-infinity. Therefore, this research focuses on developing a controller that aims at the ease of implementation, minimal computation load and flexibility to be adaptable to platform changes such as payload and sensors, at the same time being able to perform comparably well in simulation as well as the experimental flight. Besides, most flight controllers were designed for an ideal situation or validated in simulation with artificial noise added. Thus, part of this research focuses on designing a controller that is robust and guarantees flight performance under wind turbulent and severe environment.

1.3 Outline

This thesis is organised into seven chapters. The first chapter introduces the motivation, research objectives, and contribution of this research. The remaining chapters are organised as follows.

Chapter 2 presents the overview of the methods used in modelling and system identification, notably mathematical representation of system dynamics through first principle modelling and system identification in time domain. The overview of controllers design for UAV helicopter is also presented, being grouped into learning-based, linear model and non-linear model control design.

Chapter 3 presents the helicopter platform, test rig construction and the experimental setup used for data collection from outdoor flight experiment as well as control algorithm testing on test rig. Besides, an improved PID control scheme to stabilise the helicopter attitude is described for closed-loop system experiment.

Chapter 4 presents mathematical modelling through first principle modelling and simplification to obtain the helicopter attitude dynamics model. The integral-based parameter identification algorithm is presented for offline parameter identification. The extension of the integral-based algorithm is also introduced for real time application to account for time-varying model parameters as well as the disturbance. Recursive prediction error method (RPEM) based on ARMAX model is discussed and modification on the parameter estimator to obtain robust estimator is also presented.

Chapter 5 presents the results from modelling and system identification on the helicopter attitude dynamics. The results are grouped into the offline system identification

and the online parameters identification based on integral-based parameter identification algorithm. The system identification data are collected from the outdoor flight and closed-loop control experiment.

Chapter 6 presents model predictive control scheme that is based on the low-order plus dead-time nominal model. Generalised Predictive Control (GPC) is considered due to its insensitivity with respect to plant model mismatch and offset-free performances capability. In addition, certainty-equivalence adaptive control based on underlying robust GPC is presented where recursive parameter estimator based on RPEM is constantly tuning for the true nominal model in order to improve the performance of the underlying robust controller. The last part presents a model predictive control scheme that uses the iterative integral method to identify disturbance.

Chapter 7 presents the experiment results from the implementation of SP-based GPC on the yaw channel of helicopter to control the heading in hover or near-hover condition, which includes the effects of control objective design parameters tuning. The comparison of performance is made between the fix model SP-based GPC and the adaptive SP-based GPC. The experiment results from the implementation of yaw heading control using model predictive control with disturbance identification are presented at last.

Chapter 8 presents the concluding remarks for this thesis, and recommendations for future research work are also suggested.

2. LITERATURE REVIEW

2.1 Dynamics Modelling and Identification

Mathematical representation of RUAS dynamics is commonly derived from first principle theories. The resulting model is a set of differential equations and in high order system that link different subsystems together. Each subsystem is considered as a dynamic system on its own, for instance the main rotor flapping dynamics, fuselage dynamics, air inflow dynamics etc. If all subsystems are considered, the model would be very complex (Prouty, 1995). Therefore, most published papers used only the main four subsystems – rigid-body dynamics, force and torque generation mechanism, rotor aerodynamics and dynamics and actuator dynamics as shown in Figure 2. Rigid-body dynamics is generally described by Newton-Euler equations of motion, or the Lagrange formulation. The force and moment generation mechanism is not a dynamical system but a process that transforms the orientation of the thrust and torque generated by rotor to the resulting force and moment vectors experienced by the rigid body. Orientation of rotor is controlled by tilting the blade (swashplate mechanism) or the rotor itself. Detailed modelling of rotor aerodynamics and dynamics can be very complicated (Chen, 1979), thus a simplified rotor dynamics using a combination of momentum and blade elements theory is normally adopted. To improve the model fidelity, actuator dynamics such as swashplate servomotors is a necessity, due to the delay of actuator output response to the commanded input signal.

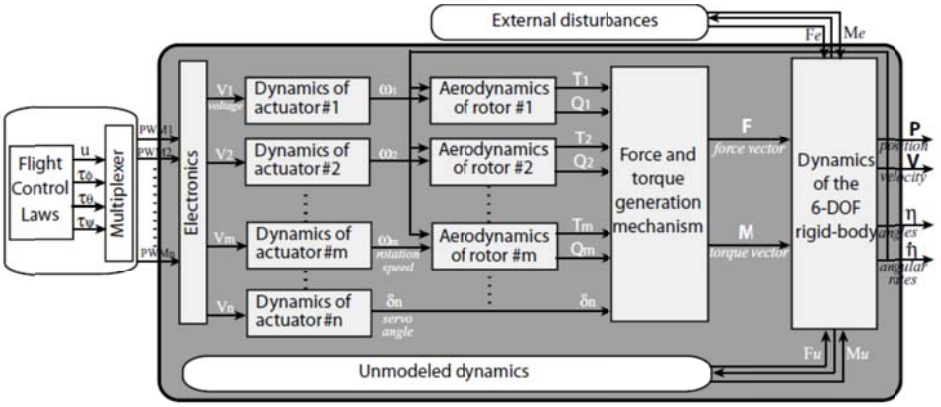


Figure 2: RC Helicopter dynamics (Kendoul, 2012).

First principles modelling typically produces simulations model of nonlinear dynamics and an extended flight envelope, which provides the capability of extracting linear models at various trim operating points (Kanade, 2002b). The first principle modelling technique derives the mathematical equations using the fundamental laws of mechanics and aerodynamics, thus it requires an extensive knowledge of helicopter dynamics and all phenomena involved in flight. In addition, the mathematical model also contains dozens of unknown physical parameters such as geometrical data and aerodynamics coefficients. Therefore, significant experimentation and tuning are needed to accurately determine the underlying physical parameters. On the other hand, the system identification technique is essentially a data-fitting process that depends on input-output data mapping to build a mathematical model, specifically linear model. Most notable example is the CIFER (Comprehensive identification from frequency responses) tool. CIFER is used in rotorcraft identification based on frequency-domain technique. As system identification method is normally constrained on linear model and hence the model lack

the expressiveness to comprehend the dynamics of rotorcraft, therefore first principle modelling can be combined with system identification technique for rotorcraft modelling. Such an approach is presented in (Kanade, 2002b) where the unknown parameters in nonlinear rotorcraft dynamics is identified through an optimization method that tunes the parameters value to match the frequency responses from flight data over multiple operating points. Similar approach is presented in (Abbeel et al., 2010b) for identifying simple nonlinear dynamics model of a small aerobatic helicopter (XCell Tempest). Furthermore, there are instances where the system identification is carried out in time-domain, such as the prediction error method (PEM) (Hyunchul Shim et al., 2000), maximum likelihood method (Bruce P.D., 1998), equation error method and output error method (Raol et al., 2004). However, these method are usually very time consuming, and sensitive to starting point; and has no guarantee of global optimality (Hanbo et al., 2009).

Early research on the identification scheme and control design for model-scaled helicopter are based on linearized dynamics (Morris et al., 1994) of the helicopter using the concept of stability derivatives (Bhandari, 2005). Flight dynamics modelling is typically broken into operating regions such as hovering or forward flight. The model is valid within a certain frequency range, where good linear correlation between angular rate and cyclic inputs is found (Tischler, 2000). Several works (Salman et al., 2006; Kallapur and Anavatti, 2006; Lyashevskiy and Yaobin, 1996) directly identify the nonlinear model dynamics using state-space identification method and Extended Kalman Filter (EKF). However, they often require good initial estimate of initial states, which may not always be available.

2.2 Learning-Based Flight Control

Over the past two decades, there has been considerable research into control design for RUAS, resulting in a large number of published papers (Ollero and Merino, 2004; Valavanis, 2008) and various control techniques for RUAS. The most commonly known techniques include learning-based flight control systems, linear flight control systems and model-based nonlinear controllers. For learning-based flight controller, the most used techniques are fuzzy logic, human-based learning, and neural networks.

Fuzzy logic based controller has proved to be effective for RUAS control. An example is a model-free fuzzy control applied to Yamaha R-50 unmanned helicopter (Sugeno, 1995). The experimental flight tests indicate that the fuzzy logic controller is sufficient for basic flight manoeuvre such as hovering, forward flight and climbing turns. The control architecture follows the typical hierarchical based design, where the lower level control module generates the control inputs to actuators for flight stabilizing while receiving commands from higher level control module that decides the flight course. The control rules in fuzzy logic control are built from the information and expert pilot knowledge. The piloted flight data is used in generating and adjusting the fuzzy rule base. Similar approach is implemented for USC AVATAR helicopter (Montgomery and Bekey, 1998) based on hybrid fuzzy logic and general regression neural network controller. The validation in simulation environment shows a promising result, but is inadequate for flight-test on actual robotic helicopter. A recent attempt of using fuzzy logic based

controller for a RC Maxi-Joker helicopter is presented (Garcia and Valavanis, 2009). Decoupled fuzzy controllers are implemented for pitch-longitudinal motion, roll-lateral motion, yaw and collective-vertical control. The methodology is validated by experimental results of over 300 autonomous flights that include hovering, take-off and landing, forward flight and waypoint flight.

Human-based learning approach is explored by the MIT researchers on acrobatic flight manoeuvring (Gavrilets, 2001). The technique is based on analyzing the input sequence that is required to execute an aggressive manoeuvre from which intuitive control logic is developed and implemented (Gavrilets, 2004). A reinforcement learning control approach is proposed in a recent published work (Abbeel et al., 2010a). The demonstration of desired manoeuvring is input to a trajectory learning algorithm to obtain a target trajectory. The output from the learning algorithm is used to learn a high accuracy dynamics model based on the flight data from desired manoeuvre. A standard linear quadratic regulator (LQR) is applied off line to find an estimate sequence of quadratic cost-to-go functions for each time until the end of trajectory. The helicopter is then run autonomously by executing an off-line LQR-based controller. The process is repeatedly refined with an improved model until a satisfactory result is obtained. This method is shown to be able to tackle the challenging control problem of performing aerobatic manoeuvre; however, the controller may not be able to account for the unexpected disturbances that drive the helicopter away from the desired trajectory.

Artificial neural networks (ANN) based controllers are presented in several works, where ANN is used to identify dynamics model and developing an ANN-based controller (Buskey et al., 2001). There are also works that use the identified dynamics model to build a standard control (Johnson, 2005).

Overall, learning-based approaches have shown promising results, however, extensive experimental evaluation over a wide range of environments and scenarios has yet to be performed. Besides, ANN based controllers especially the online identification that updating the neural network parameters at every sampling interval for control computation are computationally demanding, which impedes their applications for real time fast sampling control problems.

2.3 Linear flight control

Proportional integration derivative controller (PID) is the most widely used linear controllers. The PID-type flight controllers are implemented (Castillo et al., 2005a) and are still in use in robotic helicopters. Generally, a two-stage cascaded design is implemented. The outer loop responsible for translation motion control, whereas the inner loop controls the attitude of helicopter. Single input single output (SISO) PID controller is applied to each axis assuming decoupled dynamics. Such a controller is shown to be adequate for non-aggressive flight conditions and for rotorcraft UAVs with limited computational power.

Linear-quadratic regulator (LQR) is another popular control technique that is frequently applied to control rotorcraft UAVs (Gavrilets et al., 2002; Tanner, 2003). At Chiba University, experimental flight test on rotorcraft UAVs of different weights (2 to 48kg) demonstrated a stable hovering and accurate trajectory tracking using LQR-based cascaded controllers designed from linear models (Shin et al., 2005). In addition, an extension on standard linear LQR controller has been reported (Bogdanov A., 2003). The work presents a LQR-based controller that solves the Riccatti equation at every real time step using the continuously updated state dependent representation of the rotorcraft dynamics. This approach shows a better performance than steady-state LQ controllers based on traditional linearization. A further extension to (Bogdanov A., 2003) is presented in (Chowdhary, 2005), where the proposed LQ controller is combined with online parameter identification mechanism to account for changing system dynamics, parameters uncertainty, sensor noise and external disturbances.

The H_∞ control approach is a robust control design method that accounts for parametric uncertainty and unmodeled dynamics. It has been implemented in full scale helicopters (Smerlas, 1998). One research work on H_∞ based controller is the H_∞ loop shaping controller developed by (La Civita, 2006), which is based on linear model extracted at hover. The controller was tested on Yamaha R-50 and demonstrated a good tracking performance during a set of manoeuvres. Excluding the work by La Civita mentioned before, most of the published works mainly focus on theoretical aspect of using H_∞ for helicopter control (Gadewadikar et al., 2009; Gadewadikar et al., 2008). This is due

to the numerically and theoretically complicated optimization problem of H_∞ norm, which somewhat limits the practical control system application.

Linear controller such as those mentioned above suffers from deteriorated performance when the helicopter operates beyond the nominal conditions during aggressive manoeuvre. Therefore, nonlinear dynamics of rotorcraft UAVs is modelled as a collection of linear models that represent specific operating regimes. Linear controller can then be implemented for each linear model. Such controller is known as gain scheduling control. The same principle is adopted by the Army/NASA for developing the controller for Yamaha RMAX helicopter (Takahashi, 2008). The controller is designed based on linear models identified using CIFER software, with data collected from frequency sweeps in hover and forward flight as input to the software. Feedback gain of PID-like controller is designed to have low value at higher frequencies and high value at lower frequencies. Gain scheduling is then used to assign changing feedback gain, for example during transition from hover to forward flight.

In summary, despite the well-known limitation in operating regime, linear controller is still the most widely accepted method of flight control due to its ease of implementation, availability of many tools to design and analyze the performance and robustness. In addition, they have been successfully employed in a wide range of flight tasks and manoeuvres.

2.4 Model-based nonlinear flight control

Nonlinear flight controllers are control designs that directly address the limitation of linear controllers. Generally, these flight controllers are based on nonlinear model of the rotorcraft dynamics. A variety of nonlinear controllers have been developed and applied to rotorcraft UAVs. Among these, feedback linearization, adaptive control, model predictive, and backstepping control have received much attention and been successfully applied to helicopter control.

Feedback linearization uses nonlinear transformation technique to transform the state variable of the system into a new coordinate system, where the dynamics is linear. Standard linear tools can be applied for the control design and subsequently converted back into original coordinates via an inverse transformation. Exact input-output feedback linearization on rotorcraft dynamics is found to possess unstable zero dynamics. Thus approximate linearization is used for tracking control synthesis (Koo and Sastry, 1998) and performance is verified in simulations. A combined nonlinear feedback control of inner loop and dynamics inverse for outer loop is proposed (Peng et al., 2009). The inner loop consists of linear feedback control law for velocities, attitude, heave velocity and heading control using linear model identified from flight data. The reference velocities input to inner loop is computed by outer loop position controller using dynamics inversion technique. Flight test on Raptor 90 helicopter has shown several successful manoeuvres, such as automatic take off and landing, hovering, slithering, pirouetting, spiral turning and etc.

The performance of a feedback linearization control is limited to an accurate nonlinear model, which most of the time is hard to obtain. Therefore, robust nonlinear control scheme that can handle modelling errors and uncertainties is proposed, known as adaptive control. (Johnson, 2005) presents an adaptive controller developed and implemented for trajectory tracking on Yamaha RMAX helicopter. In that adaptive control scheme, a generic feedback linearized controller is augmented with neural-network-based adaptive control architecture that compensates for unknown nonlinearities. The neural-network is used to parameterize the unknown model error and perform online parameters adaptation to minimize the effect of parametric uncertainty. In another attempt on adaptive control of an unmanned rotorcraft, an integrated observer-controller form of adaptive control was developed for helicopter slung load operation (Bisgaard et al., 2010). The observer constantly updates the model with new position and the length of suspension system information. The observer combined with feedback and feedforward control scheme significantly reduces the slung load swing. The performance is validated through simulations and laboratory flight tests.

Model predictive control (MPC) is also referred to as receding horizon control (RHC). The principle of MPC is to obtain an optimal control sequence through minimizing the cost function over a fixed length time interval in future time horizon. The cost function is formulated, subject to constraints of the system dynamics, inputs and tracking errors. The nonlinear MPC has been successfully applied for tracking control of unmanned helicopters (Kim et al., 2002). Experimental results of waypoint navigation, a probabilistic

pursuit-evasion game and vision-based target tracking are presented in (Kim and Shim, 2003), and the flight control system has been implemented on Yamaha R-50 helicopter.

Backstepping is known for its recursive methodology, where feedback control law is constructed at every time step. The control law formulation depends on constructing Lyapunov function from nonlinear model that satisfies certain structural properties. The resulting control law ensures the stability of closed-loop control system. Rotorcraft dynamics model fulfils the structural properties that allow application of backstepping control design. Several papers (Frazzoli et al., 2000; Mahony et al., 1999; Olfati-Saber, 2001) have been published based on backstepping control for unmanned rotorcraft, however, the proposed controllers were only validated in simulation.

In conclusion, theoretical works on nonlinear controllers show that nonlinear controllers have wider operating envelopes and outperform linear techniques in terms of robustness to unmodelled dynamics and disturbances, tracking accuracy over wider flight envelopes and etc. However, very few works report experimental results that can show great improvement on flying capabilities compared with linear controllers. Thus, rigorous implementation and flight-testing on developed algorithms are required.

3. HELICOPTER RESEARCH PLATFORM DEVELOPMENT

3.1 Introduction

The research in this thesis is carried out in two phases, mainly the experiments for characterising helicopter dynamics and control algorithm development and validation. For the first experiment, a scaled model helicopter Trex 600 ESP is used for data collection in an outdoor flight. As for the control system testing, Trex 450 Sport V2 is used instead for doing indoor experiment on a test rig. Two different avionics systems have also been developed for the outdoor experiment and experiment on test rig respectively. The experiment on test rig is designed mainly to avoid any potential mishap and crashing to the ground that might have happened when the control system fails to perform up to the expectation. Thus, with the test rig the repair cost can be saved and the safety of tester can also be ensured.

This chapter is organised as follow. In Section 3.2, the RC model helicopter hardware and the working principle are described in detail. Next, in Section 3.3, the experimental set up and data acquisition are presented, and in Section 3.4 the construction of test rig and the corresponding experimental set up for control testing are elaborated in detail. Subsequently in Section 3.5, a PID controller implementation is described. The major software component is developed for helicopter stabilisation without pilot manoeuvre so that closed-loop data can be acquired for system identification. Finally, Section 3 gives a summary for this chapter.

3.2 RC Helicopter Description

Trex 600 is equipped with Bell-Hiller stabilizer bar that increases the stability of the helicopter by providing damping on pitch and roll response. With a two-bladed rotor of 0.6m radius and high torque brushless motor, it has a payload capacity of 2kg which allows it to carry the instrumentation equipment and fly for about 15 minutes. The Trex 450 Sport V2 remote control (RC) helicopter is smaller in size compared with Trex 600 SE and is chosen as the platform for carrying out experiments on test rig to demonstrate and validate the minimal modelling and system identification methods of this research. It is equipped with Bell-Hiller stabilizer bar and has a two-bladed rotor of about 0.3m radius.

The RC helicopters generate the required aerodynamics lift force and moments by changing main rotor RPM and rotor blade angle of attack to obtain the desired attitude and position control. Electronic Speed Control (ESC) governs the main rotor rotation speed and the rotor blade pitch angle is controlled using the swash plate mechanism on the main rotor hub. When both rotor blades have the same angle of attack around the rotor hub commanded by collective pitch, it creates an aerodynamics lift force. On the other hand, if one rotor blade has positive angle of attack and the other has negative angle of attack it will create a moments on the helicopter and enables the longitudinal and lateral movements (commanded by longitudinal/lateral cyclic pitch). Figure 3 shows the swash plate mechanism and the three digital servo motors that will tilt the swash plate. The three servo motors are arranged in 120 degrees around a swash plate so that each servo motor can elevate one side of swash plate and their combined efforts represent the cyclic and collective pitch input command of the helicopter.

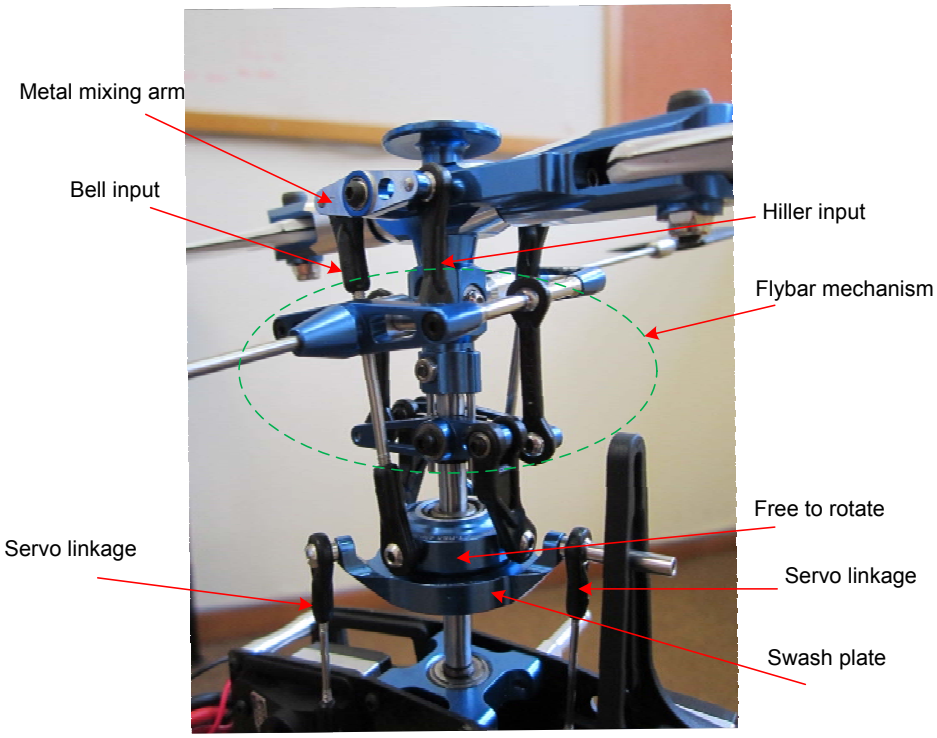


Figure 3: Swash plate mechanism and Bell-Hiller flybar mechanism.

The flybar mechanism commonly used in RC model helicopter is originated from the design concept of full-scale helicopter and is an incorporation of some design aspect of Bell and Hiller system. Figure 3 also shows the flybar mechanism. The two linkage rod coming from the swash plate known as Bell input enables tilting of the swash plate to directly affect the cyclic angle of the rotor blade. The response with respect to this rod alone is fast, but lack stability. The other two arms that come from the swash plate are each connected to the flybar control rod via a mixing arm, which allows the pitch of the flybar to change and thus creates flybar flapping angle. This flapping motion of flybar causes the cyclic angle of the main rotor blade to change via Hiller input. The effective actuator force to change the rotor blade angle is then the combination of Bell and Hiller input through a

metal mixing arm. Thus, the flybar can be considered as the secondary main rotor blade angle actuator and eliminate the need for large control forces on the cyclic servo actuators.

On the other hand, in the presence of aerodynamics forces and external moments, the helicopter pitch and roll angles θ, ϕ are upset, but the flybar tends to remain in the same plane of rotation due to gyroscopic effect and essentially changing the flapping angle to a non-zero value, as shown in Figure 4. An appropriate compensation input is then automatically created from the Hiller input to the main blade cyclic control system to stabilise the helicopter.

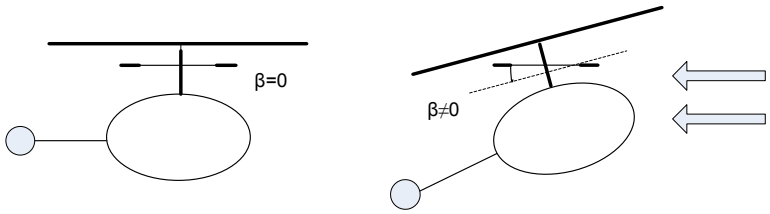


Figure 4: In hovering situation, the flybar flapping angle is zero. If a wind gust knocks the helicopter out of its equilibrium, the flybar maintains its rotation in the same inertial plane and the flapping angle become nonzero. Adapted from (Kim and Tilbury, 2004).

The main purpose of tail rotor is to counteract the reactive torque caused by the torque of main rotor. The force generated from the tail rotor is determined by the collective pitch setting of the tail rotor blade and is directly actuated by a single servo motor. The tail rotor is also used to change the helicopter heading. Due to the small size of RC model helicopter, the torque associated with the yaw channel is highly sensitive. As a result a small amplitude input and wind gust disturbance may produce a large change in yaw rate. Thus, to make the manual hovering of hobby helicopter easier, the helicopter is commonly

equipped with a yaw rate gyro, which consists of a low-cost yaw-angular-rate sensor and a simple controller to stabilise the yaw rate. In the event of cross wind while the helicopter is in hovering mode or when the helicopter in high speed flight, the gyro prevents the drift in yaw axis by computing the drift angle and constantly outputs a control signal that resists the cross wind or reactive force from main rotor torque. In other words, the gyro automatically corrects (auto trim) changes in helicopter tail trim.

3.3 Flight Instrumentation Setup for Outdoor Flight Experiment

In order to record the data for system identification, data logging is implemented through an onboard Mobisense MBS270 embedded computer. MBS270 runs as a standalone computer with kernel and file system in flash that support Linux OS. Thus, it offers faster software development since the program can be developed in the high-level programming language in C/C++ with supplied open source libraries. The Mobisense MBS270 is the central processing unit in our application which interfaces with an Inertial Measurement Unit (IMU) from MicroStrain 3DGX1 via RS232 serial interface to log the helicopter attitude information. The IMU unit performs the filtering and processing on the raw sensory output data from accelerometer, gyroscopes, and magnetometers to output linear acceleration, angular rates and orientation of the helicopter.

A separate microcontroller Arduino Duemilanove is also used for reading the helicopter actuation inputs, which are the pulse width modulated signals for four servomotors. The four servomotors are the actuation input to the helicopter. Three of them are arranged in 120 degrees around swash plate so that each servo motor can elevate one

side of swash plate and their combined effort enables cyclic and collective pitch on main rotor blade. The last servomotor is for applying pedal input to the helicopter by adjusting the tail rotor blade pitch angle. The recorded servo signals are then transferred to the MBS270 through Universal Asynchronous Receiver/Transmitter (UARTs) line.

The input output data collected during flight test is stored into a microSD memory card on MBS270 as a text file and then transferred to a PC workstation for analysis after landing. The overall architecture of the data acquisition system is shown in Figure 5.

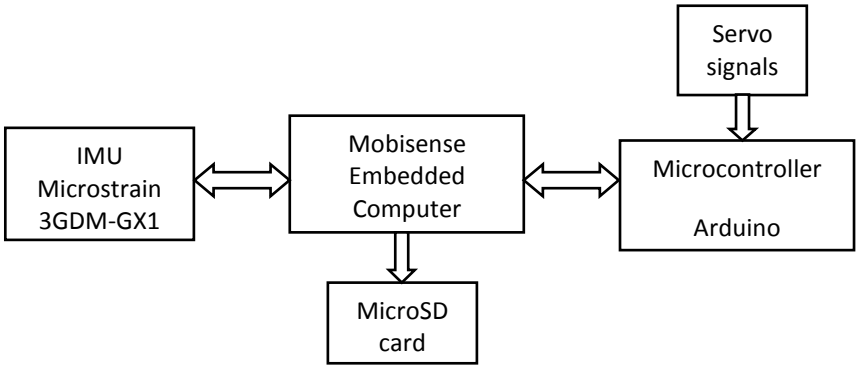


Figure 5: Architecture of Data Acquisition System

In order to correlate the control inputs to the four servos pulse width modulation (PWM) signal, an experiment is performed considering the linear correlation:

$$\begin{bmatrix} U_{AILE} \\ U_{AUX} \\ U_{ELEV} \\ U_{RUDD} \end{bmatrix} = G * \begin{bmatrix} \delta_\phi \\ \delta_\theta \\ \delta_o \\ \delta_\psi \end{bmatrix} \tag{3.1}$$

where $\begin{bmatrix} U_{AILE} & U_{AUX} & U_{ELEV} & U_{RUDD} \end{bmatrix}^T$ are the four servo motor signals pulse width measurement, $\begin{bmatrix} \delta_\phi & \delta_\theta & \delta_o & \delta_\psi \end{bmatrix}^T$ are the normalised pilot sticks range (± 1 for $\delta_\phi, \delta_\theta, \delta_\psi$

and δ_o). After substituting the obtained data from a few combinations of control input with a corresponding PWM signal, the resulting formula is obtained:

$$\mathbf{U} = \mathbf{G}'^{*}\delta + \mathbf{U}_{trim} \tag{3.2}$$

where $\mathbf{U}_{trim}$ is the pulse width for starting position of pilot stick position, $\mathbf{G}'$ is the determined control input gain. In order to access the control input at any time, the following equation translates the servo motors pulse width to the corresponding normalised control input:

$$\delta = \mathbf{G}'^{-1}(\mathbf{U} - \mathbf{U}_{trim}) \tag{3.3}$$

3.4 Development of Test Stand for Flight Control Test

A test rig was built for the RC helicopter in order for indoor experiments to be carried out in a safe manner and to allow individual axes in the helicopter to be isolated (Weilenmann and Geering, 1994). It also allows validation of the parameter identification method without the significant disturbance from the wind indoors, and is thus an intermediate step to bridge the gap to full flight. Figure 6 shows the helicopter test set up with the test rig. The test rig was built to have 4 degrees of freedom (DOF) which are the three attitude axes and one vertical movement. The pitch and roll angle on the test rig are limited to -40 to 40*deg*, while the maximum vertical displacement is 40*cm*.

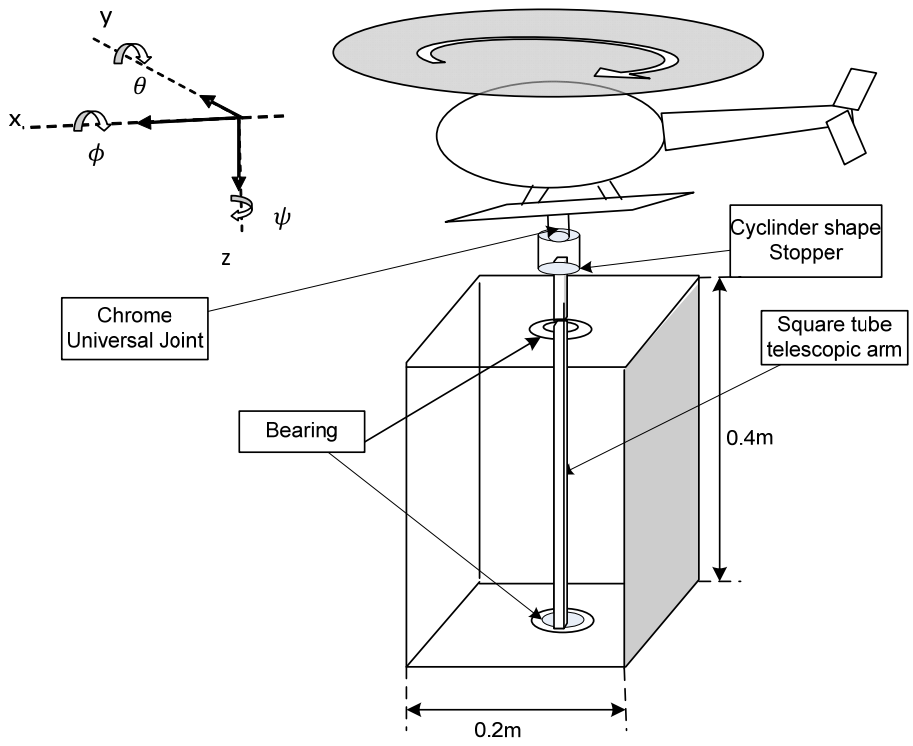


Figure 6: Helicopter test rig.

In order for helicopter to spool up from the ground level, a servo mechanism is built so that certain main rotor rotation speed could be reached before the restraint on the pitch and roll axis of helicopter is released and let the automatic control to take over the task of helicopter stabilisation. This extra setup allows the main rotor to attain enough power to perform the roll or pitch control action so that the control command from the controller could be performed instantaneously. A golden spiral cam is fabricated from Perspex and the central axis is attached to a servo motor. A tangent surface is elongated from the second quarter of the cam to form a flat surface for holding the helicopter in ground level. The mechanism is shown in Figure 7.

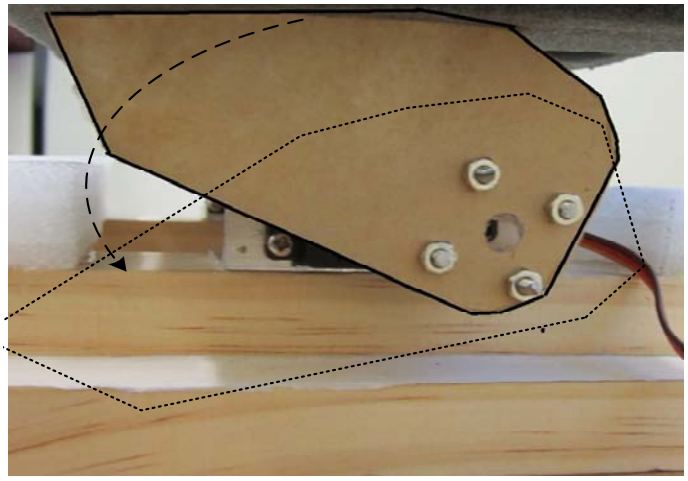


Figure 7: Mechanism for level positioning of helicopter.

3.4.1 Flight Instrumentation Setup for Automatic Control Test

An onboard microcontroller Arduino Due board is used to interface with a low cost 9 Degree Of Freedom(DOF) razor Inertial Measurement Unit (IMU) and the four servo actuators. The Arduino board is based on a 32-bit ARM core microcontroller with 84MHz clock, 54 digital input/output pins, 4 hardware UARTs (hardware serial) for TTL serial communication and etc. Both the IMU reading rate and the servo actuators updating frequency are set at 50*Hz*. However, the control signal computation is not done on board, instead it depends on the ground station to provide the appropriate control command either for tracking a reference or for regulation purpose. This kind of setup is mainly to lighten the payload on the helicopter and at the same time the control algorithm can be run in high speed using a faster computer. Besides, the reference trajectories can also easily be altered on the fly from ground station. In order to have a wireless transmission between

onboard microcontroller and ground station, Bluetooth is used for communication between the onboard microcontroller board and a laptop that runs the control algorithm. Once the communication is established between the onboard microcontroller via a Bluetooth transceiver modem BlueSMiRF Silver and laptop with built-in Bluetooth devices, the IMU data and servo command signals can be sent to and received from the laptop wirelessly within a distance of approximately 18m. Figure 8 shows the overall setup.

The microcontroller board acts as a relay for information passing between helicopter and host laptop via Bluetooth link, which enables control algorithm implementation and data logging for system identification to be done on the host laptop. Besides, the microcontroller board has to constantly measuring the positive pulse width of servo signal from radio receiver, specifically for the GEAR channel that acts as a switch for switching between pilot control and software control. If it is under pilot control, all the four servo channels, i.e. AILE,AUX1,ELEV and RUDDER are read from the radio receiver. On the other hand, if it is software control, the servo signals will be obtained from the host laptop via Bluetooth communication. In order for efficient pulse width measurement of the servo signal from the radio receiver, interrupt is used in the controller for triggering an event whenever there is a change in the pulse level on a port pin. The flow chart of the software in Arduino Due is shown in Figure 9.

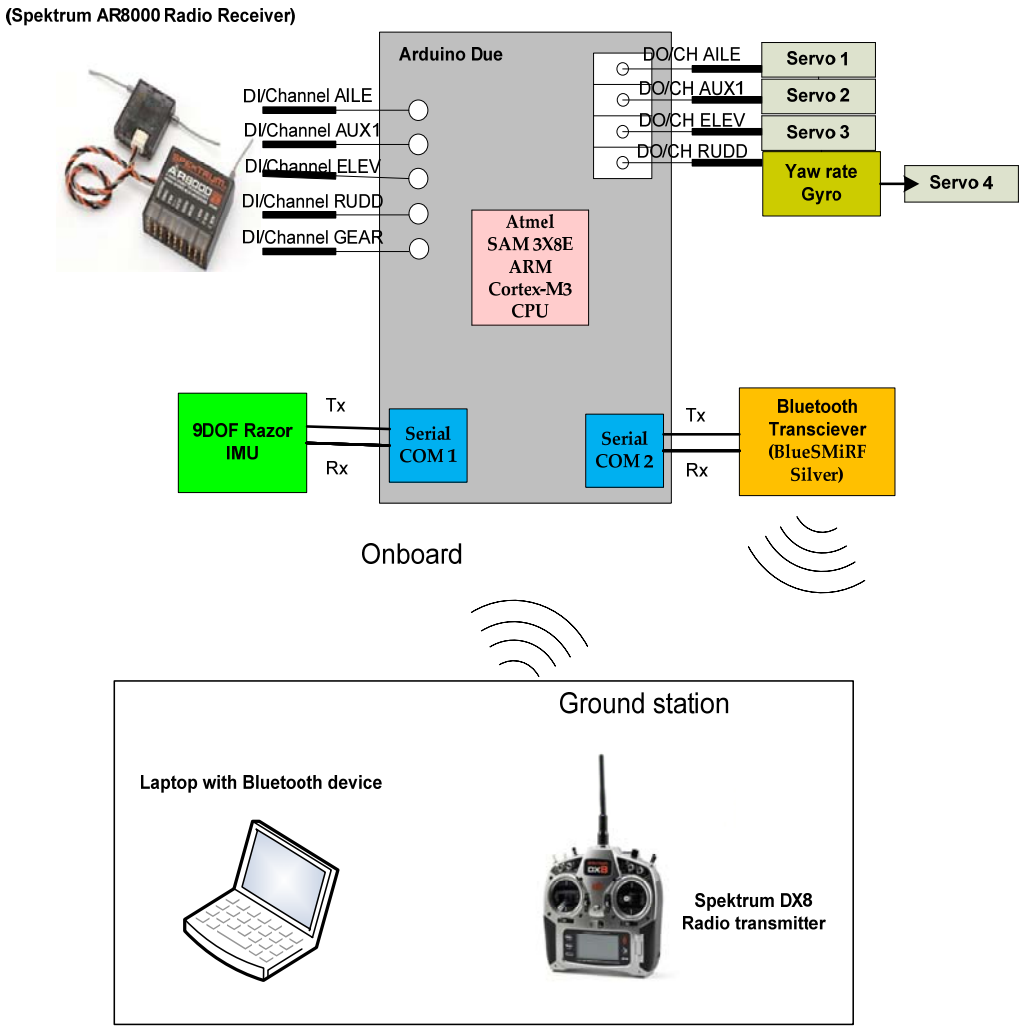


Figure 8: Instrumentation setup on test rig.

The 9-DOF Razor IMU is used as the attitude and heading sensing device on board of the helicopter. It is mainly composed of three sensors which are the ITG-3200 (triple-axis gyro), ADXL345 (triple-axis accelerometer), and HMC5883L (triple-axis magnetometer). Although the output from the accelerometer and magnetometer can be used to obtain the angular information and the heading respectively, the accelerometer is sensitive to linear acceleration and local gravitational field. In the present of translational movement, the rotated gravitational field will be interfered by the linear acceleration and results in inaccuracy of pitch and roll orientation angles.

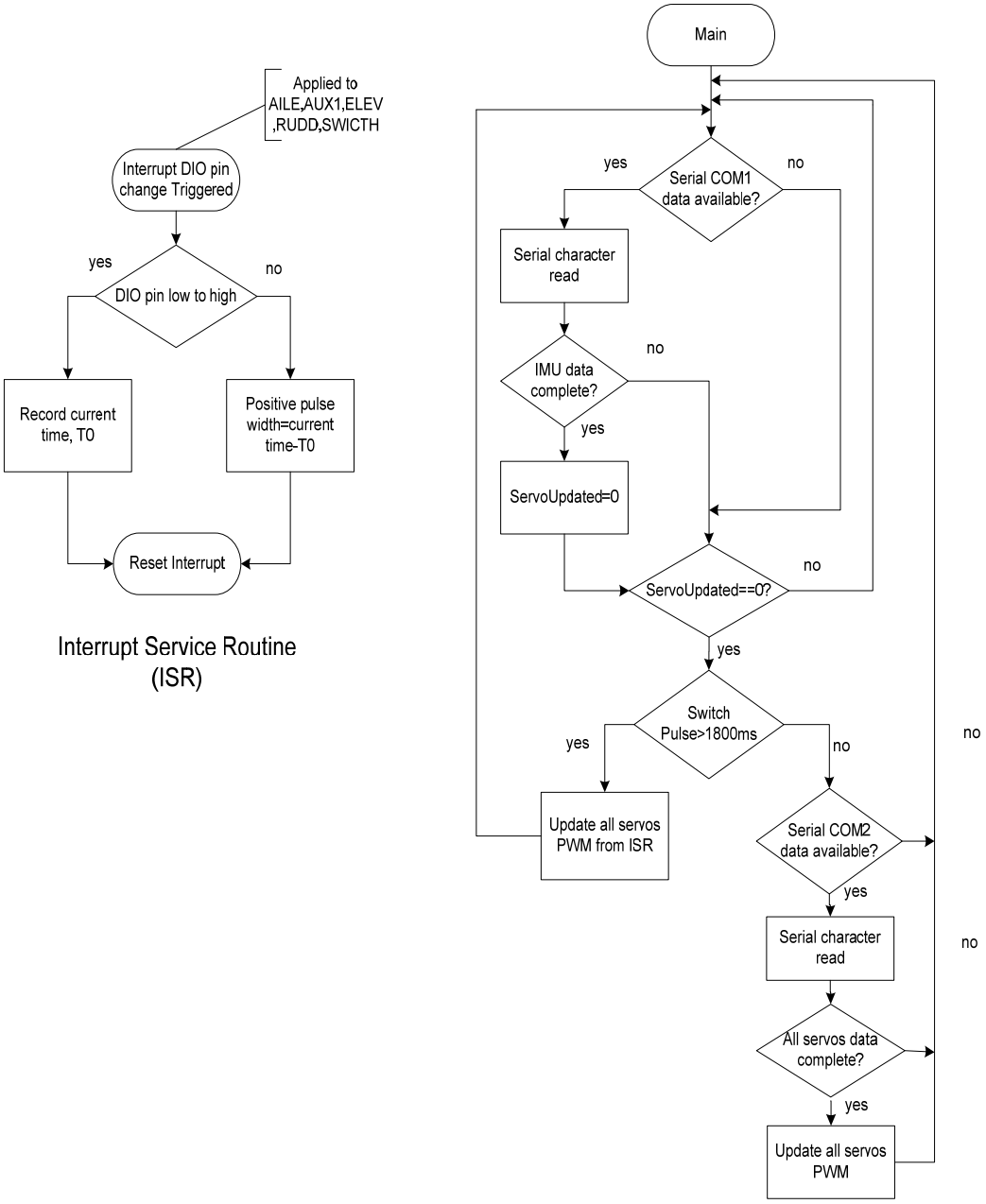


Figure 9: Flow chart of the software in onboard Arduino Due

In addition, accelerometer is also sensitive to high frequency noise. On the other

hand, integration of the velocity from the gyro can generate angular rate and angle information. However, since the angle is not an absolute angle, reference is needed. Besides, due to time-varying small offset in the angular rate, there is a drift in the generated angle over time. As a countermeasure, the 9-DOF inertial measurements from the three sensors have to work in a complementary mode and are fused by using the Direction Cosine Matrix (DCM) algorithm, which runs in the ATmega 328 microcontroller. The resultant measurements are sent over the serial interface every 20*ms* or 50Hz to the Arduino Due board. The measurements data are 3 axes (roll, pitch and yaw) angles and angular rates of body frame with respect to fixed earth frame. The helicopter body axes are as shown in Figure 10.

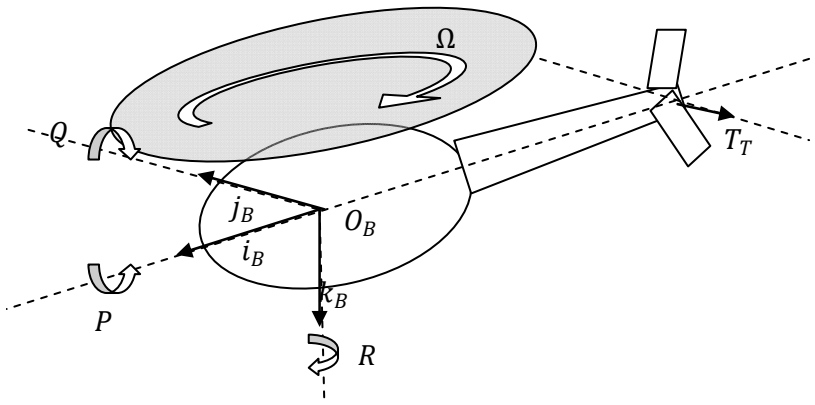


Figure 10: Six degree of freedom of helicopter in body frame

3.5 Implementation of Proportional Derivation Integration (PID) Control

Due to inherent instability of helicopter, and for the economic or safety reasons, data collection for system identification is performed in a closed loop. Closed-loop experiment means that an extra feedback controller is included to ensure the stabilisation, and the feasibility and identifiability have been discussed in (Ljung, 1987). Compared with the open-loop experiment, where manual control is required, the input excitation can be made richer in information and specifically over interested frequency range, which can be difficult to generate by manual control. Besides, since the designed controller is known, the closed-loop identification can be transferred to open loop identification and open-loop identification algorithm can be used. PID controller is deployed in closed-loop experiment for data collection, which is in turn used for system identification. PID controller is commonly employed due to its simple feedback algorithm, reasonable closed loop performance, and simple implementation on digital computer. In addition, it can also be used as a comparison with model based control, which will be discussed in Section 6.

A PID controller is the summation of proportional, integral and derivative action on the control error. The proportional action generates the counter error on the control error. The integral action is related to past values of control error and making the steady state error to converge to zero when a step a reference signal is applied or a constant load disturbance q occurs. The derivative action depends on predicted future value of control error, thus can improve the performance by anticipation of the trend of control error and

counteract for it. The structure of the PID scheme is shown in Figure 11, where $P(s)$ is the process model. The control action is expressed as

$$u(t) = K_p(e(t) + \frac{1}{T_i}\int_0^t e(v)dv + T_d\frac{de(t)}{dt}) \tag{3.4}$$

where $e(t)$ is the control error, K_p is the proportional gain, T_i and T_d are the time constants of integrative and derivative terms respectively.

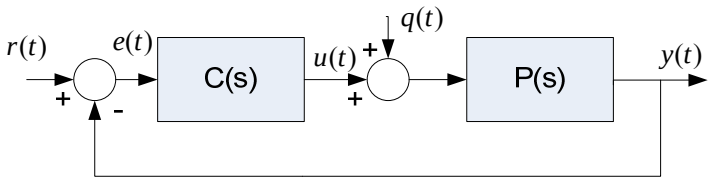


Figure 11: Standard unity feedback control scheme.

3.5.1 Implementation Considerations

The corresponding transfer function of (3.4) is

$$C(s) = \frac{U(s)}{E(s)} = K_p(1 + \frac{1}{T_i s} + T_d s) \tag{3.5}$$

The transfer function in (3.5) is improper and a filter is usually added to the derivative term such that

$$C(s) = K_p(1 + \frac{1}{T_i s} + \frac{T_d s}{1 + \frac{T_d s}{N}}) \tag{3.6}$$

or alternatively

$$C(s) = K_p(1 + \frac{1}{T_i s} + T_d s)\frac{1}{T_f s + 1} \tag{3.7}$$

The second option is employed in this research where additional IIR Butterworth filter of $5Hz$ cut-off frequency is implemented. In addition, high frequency measurement noise can also be eliminated by the filter. This is important as PID closed loop performance is very sensitive to measurement noise.

When tuning the controller parameters, there is a trade off between disturbance rejection and set-point responses performance. Normally, when the PID is tuned to obtain fast disturbance rejection without oscillation and small overshoot, a high overshoot will be observed in the set-point responses. It is caused by the zeros of the PID controller that only appear at the transfer function $Y(s)/R(s)$, which has the effect of speeding up the step response (Normey-Rico, 2007). Therefore, a two-degree-of-freedom (2-DOF) PID structure is to be used to decouple the tuning for disturbance rejection and set-point tracking performance. This is achieved through using an independent reference filter as shown in Figure 12. $F(s)$ is computed as follows:

$$F(s) = \frac{1 + bT_i s + cT_i T_d s^2}{1 + T_i s + T_i T_d s^2} \tag{3.8}$$

where b and c are the set-point weightings in proportional and derivative action respectively. From Figure 12, it can be seen that the parameters b and c do not affect the transfer function relating the disturbance and the output, $Y(s)/Q(s)$. In fact, the parameters b and c are selected to change zeros of the final $Y(s)/R(s)$ that will determine the set-point tracking performance. c is often set to zero, indicating that the derivation action is applied to the output response instead of control error, so that an impulse in the control signal is avoided when significant changes occur in set-point such as step signal.

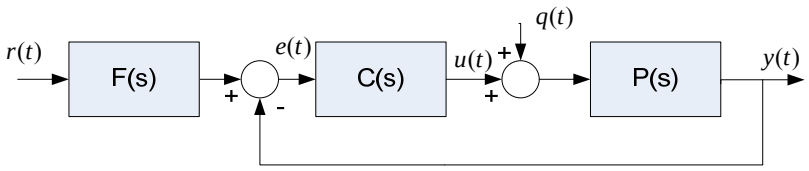


Figure 12: 2-DOF PID.

3.5.2 PID for Control of Dead-Time System

PID control can be interpreted as predictive controller and used to control process with dead-time. The derivative action allows prediction of the future trend of the control error, while proportional (Integrative) action is to satisfy the closed loop performance specification. Considering the PD action of the form

$$u_{PD}(t) = K_c[e(t) + T_d \frac{de(t)}{dt}] = K_c \hat{e}(t + T_d \mid t) \tag{3.9}$$

It shows that the control action can be viewed as linear prediction of the error in $t + T_d$ using only error at sampling instant t. In addition, the tuning of the derivative term is indirectly related to the dead time of the process. The corresponding structure is shown in Figure 13. If the prediction error $\hat{e}(t + T_D) \approx e(t + T_D)$, then the proportional control can be tuned as if the process is dead time free. This condition is fulfilled if the variation of error is smooth in the interval $[t \;\; t + T_D]$. When the process contains a long dead time, the prediction through derivative is not suitable, as the measurement signal does not contain enough information to estimate the future variations.

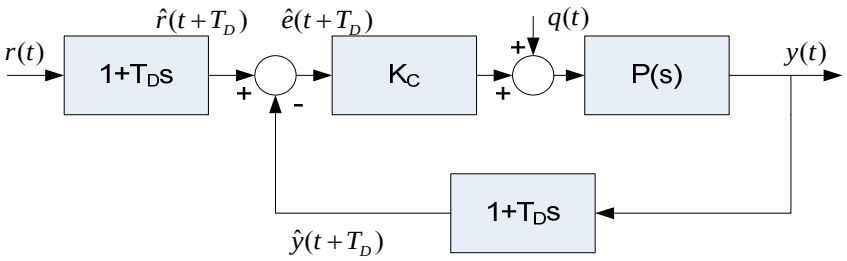


Figure 13: Prediction in PD controller.

3.5.3 Limitation of PID

Oscillatory response often observed in PID control of system with dead time and the gain of the PID control has to be reduced to eliminate the oscillation at the price of poorer performance (Normey-Rico, 2007). This can be explained using frequency domain approach where the reduction of PID gain results in smaller cross-over frequency that increases the phase margin and improve closed loop stability. However, the new value also gives a slower transient, which is the result of the phase introduced by the dead time. In addition, it is also illustrated in (Bahill, 1983) that in order to avoid closed loop control instability large gain cannot be used in a system with a large time delay relative to the time constant. Hence, desired control performance is hard to achieve when the tuning of the closed loop gain is constrained to limited set.

The derivative action of PID can be tuned to advance the phase of the open loop transfer function close to the cross-over frequency, which will directly increase the phase margin, and hence ensure well damped closed-loop response and system stability. However, if a faster closed-loop response is specified or a process with a higher dead-time is considered, it is not sufficient to obtain the desired closed-loop performance (Normey-

Rico, 2007). Therefore, there is no way to take care of both the closed loop response and stability at the same time with a PID controller.

Nevertheless, PID appears to be a suitable basic controller to stabilise or perform reference tracking for which collection of input-output interaction data for system identification can be done, specifically for unstable system or integrative system where open loop control would be difficult.

3.6 Summary

In this chapter, the RC model helicopter that is used as the research platform is introduced in terms of the mechanical structure and the electronic equipment. Then, the avionics system set up for the outdoor flight experiment is described, which is mainly for data collection for system identification. Subsequently, the construction of 4 DOF test rig specifically for consideration of safety and economic reasons is presented, and the set up for control system testing on test rig is also described. The improved PID control implementation is also presented as part of the software component for closed-loop flight experiment on test rig. The basic control also negates the need for skilled pilot to stabilise the helicopter and generating an excitation input that stimulates the characteristic response from the helicopter.

4. SYSTEM IDENTIFICATION ON HELICOPTER ATTITUDE DYNAMICS

4.1 Introduction

System identification is the field of modelling dynamic systems using experimental input and output data. In terms of constructing a dynamic model that relates the input control to the system response, it can be either a model depicted in a graph of the input-to-output response or a complex set differential equations of motion (Tischler and Remple, 2006b). For the former model, system identification is carried out based on a measured set of input-output flight test data captured with necessary excitation inputs that fully characterize the system dynamics. The latter model is considered a parametric model, represented as a series of differential equations in state-space form, given as below,

$$\begin{aligned} \dot{x} &= Ax + Bu\left(t-\tau\right) \\ y &= Cx + Du\left(t-\tau\right) \end{aligned} \tag{4.1}$$

System identification determines the matrices A, B, C and D. It also plays a role in control design besides determining simulation model. It can be done parametrically, or in a black-box non-parametric form. One way to do non-parametric system identification is by using neural networks. In particular, (Leontaritis and Billings, 1985) shows that a wide class of discreet-time non-linear systems can be represented by the general, non-parametric equation:

$$y(t) = f\left(y(t-1),\ldots y\left(t-n_{y)}\right),u(t-1),\ldots,u\left(t-n_u\right),e(t-1),\ldots,e\left(t-n_e\right)\right)+e(t) \tag{4.2}$$

where y(t), u(t) and e(t) are the system output, input and noise respectively; n_y, n_u,and n_e are the maximum lags in output, input and noise respectively; $f(\cdot)$ is the non-linear

function. The model (4.2) is often referred to as a NARMAX (Nonlinear Auto Regression Moving Average with Exogenous Input) model due to its resemblance to the linear ARMAX model. Neural networks are then employed to model this non-linear function by using input-output pairs of data collected from the real system to train the neural network (Helicopter related). Neural network based system identification is basically a black box modelling approach, where the model does not carry any physical meaning in the estimated parameters. In addition, training of the neural network has to cover all possible different situations to fully describe the system dynamics. The optimizing algorithm is also computationally intensive.

To have a better understanding of the overall system dynamics, parametric modelling is favoured over non-parametric modelling (Underwood and Husain, 2010; Moreno et al., 2009). An identification experiment is performed by exciting the input with signal such as a step, a sinusoid or random signal, followed by observation of the input and output over a time interval. The recorded input-output sequences are then fitted to a parametric model of the dynamics system. This type of system identification is known as grey box identification. This type of identification not only gives physical insight of the dynamic system, the often unknown parameters in mathematical model can be estimated from the experimental data using some statistically based method. Linear mathematical model can be used if the control is only considered in certain working point, and non-linear model can also be chosen if the dynamic system exhibits severe nonlinearities or when a larger operating envelope is required.

Early research works on the identification scheme and control design for a model-scaled helicopter are based on linearized dynamics of the helicopter using the concept of stability derivatives (Morris et al., 1994). Flight dynamics modelling is typically broken into operating regions such as hovering or forward flight. The model is valid within a certain frequency range, where good linear correlation between angular rate and cyclic inputs is found (Mettler et al., 2000). A few papers (Salman et al., 2006; Kallapur and Anavatti, 2006; Lyashevskiy and Yaobin, 1996) directly identify the nonlinear model dynamics using state-space identification method and Extended Kalman Filter (EKF). However, they often require good initial estimates of states which may not always be available.

First principles modelling typically produces a nonlinear dynamic model and an extended flight envelope, which provides the capability of extracting linear models at various trim operating points (Kanade, 2002a). The main drawback of this approach is that extensive knowledge of helicopter dynamics is required. Furthermore significant experimentation and tuning is needed to accurately determine the underlying physical parameters. To circumvent the difficulties, system identification (Tischler and Remple, 2006a) can be combined with first principles modelling to identify the unknown or uncertain physical parameters. Typical approaches of system identification in time-domain are the prediction error method (PEM) (Hyunchul Shim et al., 2000), maximum likelihood method (Bruce P.D., 1998), equation error method and output error method (Raol et al., 2004). However, these method are usually very time consuming and sensitive to starting point; and they do not guarantee global optimality (Hanbo et al., 2009).

In this research, a simplified model for pitch, roll and yaw dynamics is derived from a complex nonlinear model. It is found that a fully decoupled model in pitch, roll and yaw is able to capture data equally well compared with a more complex model with state coupling. Even though the complex model is theoretically more accurate, the significant wind disturbance and other unknown dynamics dominate. Thus, the simplified model is much more suitable given significantly less computational requirement and its ease of analysis. In addition, a major advantage of a simpler model is the potential for real time implementation, which is a key motivation for the methods developed in this research.

Specifically, changing wind conditions will change the angle of attack and thus the centre of pressure of actuation surfaces. Hence important parameters like torque constants and damping associated with the attitude dynamics change over time. There may also be unmodelled flow disturbances that significantly change the helicopter dynamics compared with a calm day or an indoor environment. Therefore the research is focussed on robust minimal modelling methods in difficult environments to ensure the methods developed are extendable to all types of atmospheric conditions during flight.

An integral based parameter identification method (Hann, 2005; Hann, 2008; Hann, 2012) which was published in the biomedical field is significantly extended to account for disturbance and modelling error. Specifically, the methods published in the biomedical field are for open loop diagnosis with a low sampling rate and are not designed for control applications where the sampling rate is very high and there are significant disturbances. The integral method is also compared against standard non-linear regression to test the computational efficiency and accuracy.

Since outdoor flight of a RC helicopter is constantly influenced by wind disturbance and complex flows across the helicopter, model based control design that depends on a static model helicopter would not perform well without accounting for the changing parameters of the underlying model as well as disturbance. Current methods that have been used to estimate the parameters of helicopter dynamics on line heavily rely on offline helicopter dynamics as a starting point. For instance, the neural network based online parameters estimation (Kenné et al., 2006) that relies heavily on offline estimation of the parameters value as initial condition. Besides, on-line training of neural network is computationally expensive, thus has limited use in real-time. In addition, it still has the issue of requiring reasonable convergence time, which is not suitable for fast transient periods that would occur during windy and turbulent conditions. Another popular method is based on recursive least square (RLS) (Raptis et al., 2009a), of which the performance depends on the persistency of excitation signal to avoid windup in covariance matrix that causes the loss of tracking capability (Vahidi et al., 2005). This motivates the extension of integral based parameter identification, i.e. iterative approach of integral-based parameter identification.

From the perspective of adaptive control scheme where the digital controller is adjusted in real time, recursive parameters identification design is the major component. This motivates the study of recursive parameters identification algorithm. The model used is in discrete form and the identification algorithm is on a sample-by-sample basis. Even though RLS is one of the most commonly used recursive parameter identification methods, the assumption of white noise for the noise model leads to bias parameters estimation

when the actual noise is not white noise. On the other hand, algorithms such as Pseudo Linear Regression (PLR) have convergence uncertainty when the true system does not belong to the model set. Therefore, recursive prediction error method (RPEM) is studied and adopted as the online parameter identification for adaptive control scheme due to its convergence to the best approximation of the system, transient and finite sample properties as illustrated in (Ljung and Söderström, 1985).

This chapter is organized as follows. In Section 4.2, the model-scaled helicopter attitude dynamics model structure is presented. Section 4.3 describes the integral-based parameter identification method and Section 4.4 presents the recursive parameter identification. In Section 4.5, the summary for this chapter is given.

4.2 Helicopter Attitude Dynamics Model

An accurate helicopter model can be formed by different subsystems, such as servo actuator dynamics, rotary wing dynamics (Ahmed et al., 2009), force and moment generation processes (Prouty, 1990), and rigid body dynamics (Koo and Sastry, 1998) to become a higher order coupled nonlinear system. However, in accordance to the research objective of using simple model for modelling the helicopter dynamics and for control design, only rigid-body model is considered. The rigid body equation of motion is based on two reference frames, which are the inertial frame (subscript *I*) and the body fixed reference frame (subscript *B*). The standard equation describing angular velocity in the body frame is defined as:

$$\dot{\boldsymbol{\omega}}^{B} = \mathbf{I}^{-1}(\mathbf{I}\boldsymbol{\omega}^{B} \times \boldsymbol{\omega}^{B}) + \mathbf{I}^{-1}\boldsymbol{\tau}^{B} \tag{4.3}$$

$$\boldsymbol{\omega}^{B} = [p \quad q \quad r]^{T} \tag{4.4}$$

$$\boldsymbol{\tau}^{B} = [M_{\phi} \quad M_{\theta} \quad M_{\psi}] \tag{4.5}$$

$$\mathbf{I} = \begin{pmatrix} I_{xx} & 0 & -I_{xz} \\ 0 & I_{yy} & 0 \\ -I_{xz} & 0 & I_{zz} \end{pmatrix} \tag{4.6}$$

where $\boldsymbol{\omega}^{B}$ is the angular velocity in body fixed reference frame, $\boldsymbol{\tau}^{B}$ is the moment components along body axes, and $\mathbf{I}$ is the fuselage inertial matrix in body coordinates. Due to the symmetry of the helicopter with respect to the $x_B - z_B$ plane, the terms I_{xy} and I_{yz} are zero. Although I_{xz} is non-zero, but the value is typically much smaller than the other terms, thus it will be ignored in the model. After some analytical manipulation, Equation (4.3) can be broken down into:

$$\begin{aligned} \dot{p} &= \frac{(I_{yy} - I_{zz})}{I_{xx}} qr + \frac{1}{I_{xx}} M_{\phi} \\ \dot{q} &= \frac{(I_{zz} - I_{xx})}{I_{yy}} pr + \frac{1}{I_{yy}} M_{\theta} \\ \dot{r} &= \frac{(I_{xx} - I_{yy})}{I_{zz}} pq + \frac{1}{I_{zz}} M_{\psi} \end{aligned} \tag{4.7}$$

where p, q, r are the helicopter's roll rate, pitch rate and yaw rate respectively, and $M_{\phi}, M_{\theta}, M_{\psi}$ are the roll moment, pitch moment and yaw moment respectively.

The external moment acting on the helicopter are mainly contributed by the main rotor. By varying the angle of attack of the main rotor blade or cyclic pitch angle, an aerodynamic lift force is created. The pitch and roll moment of the helicopter are generated

through the difference in the lift force in lateral and longitudinal axes. The equations for the moment M_ϕ and M_θ in Equation (4.7) are defined as:

$$\begin{bmatrix} M_\phi \\ M_\theta \end{bmatrix} = \frac{n}{2\pi}\int_0^{2\pi}\int_0^{BR} l \begin{bmatrix} \sin(\zeta + \frac{\pi}{2}) \\ -\cos(\zeta + \frac{\pi}{2}) \end{bmatrix} dL_m d\xi \tag{4.8}$$

$$dL_m = \frac{1}{2}\rho(\Omega l)^2 ac[\alpha_3(\frac{L_3}{L_1(L_2+L_3)}\delta_{cyc} + \frac{L_2 L_4}{L_1(L_2+L_3)}\beta) - \alpha_4 \frac{\omega_{M2}}{\Omega}] \tag{4.9}$$

where α_3, α_4 is a correction factor to compensate for simplified aerodynamics, $L_{1,2,3,4,}$ are the linkage lengths in rotor hub assembly, ρ is the air density, Ω is the main rotor angular velocity, l is the position along the main rotor blade, and a is the main rotor lift slope, c is the main blade chord length and dL_m is the main rotor aerodynamics lift element. Equation (4.8) indicates that the pitch and roll moments depend on the pitch and roll input command through δ_{cyc} as well as on the flybar flapping angle β. The last term in Equation (4.9) is due to the gyroscopic effect of the rotating blade, which is the ratio between the second element of the angular velocity vector of main rotor blade ω_{M2} and main rotor rotational speed (Padfield, 2007).

For a full-scale helicopter, the stability is achieved by having bigger body size and the flapping mechanism on the rotor hub. Flapping dynamics are commonly modelled in the literature (Raptis et al., 2009b; Castillo et al., 2005b). However, RC helicopters have a hingeless rotor hub and are equipped with a flybar to increase damping on lateral and longitudinal flight motion, and enhance the stability of the helicopter. Therefore, this paper specifically chose the flybar dynamics instead of flapping dynamics as the base model for

demonstrating that the simplified models can perform equally well to complex model in the presence of disturbances during flight. Preliminary investigation showed that since the thrust vector is normal to the main rotor plane, whenever a flapping angle is present, the cyclic collective input generates a moment on helicopter attitude dynamics. Therefore, a cyclic collective term is investigated to see if it improves prediction of attitude dynamics as compared to simpler models that lump this effect into other parameters.

The Flybar flapping angle derivation starts from defining dynamics of flybar in the Euler equation (Kim and Tilbury, 2004):

$$\boldsymbol{\tau}_F = \boldsymbol{\omega}_F \times \mathbf{I}_F \boldsymbol{\omega}_F + \mathbf{I}_F \dot{\boldsymbol{\omega}}_F \tag{4.10}$$

$$\begin{bmatrix} \tau_{F1} \\ n\int_{R_1}^{BR_2} r dL \\ \tau_{F3} \end{bmatrix} = \begin{bmatrix} 0 \\ -I_f \omega_{F1} \omega_{F3} \\ I_f \omega_{F1} \omega_{F2} \end{bmatrix} + \begin{bmatrix} 0 \\ I_f \dot{\omega}_{F2} \\ I_f \dot{\omega}_{F3} \end{bmatrix} \tag{4.11}$$

where I_f is the unified rotational inertial of the flybar. The parameters $\tau_{F1,}\tau_{F2},\tau_{F3}$ are the external moment applied to flybar, which can be obtained by integrated lift elements *dL* along the length of flybar and $\omega_{F1},\omega_{F2},\omega_{F3}$ are the angular velocities of flybar around helicopter body axes.

After defining the flybar angular velocity, inertial and external forces, the flybar flapping angle is defined (Kim and Tilbury, 2004) :

$$\begin{aligned} \beta = \beta_o & \left(\frac{\alpha_1 L_7 L_8}{L_5 L_6 L_9} (-\delta_\theta \cos\xi - \delta_\phi \sin\xi) + \frac{\alpha_2}{\Omega} (\dot{\phi}\cos\xi + \dot{\theta}\sin\xi) \right) \\ & + \frac{2}{\Omega} (\dot{\phi}\sin\xi - \dot{\theta}\cos\xi), \end{aligned} \tag{4.12}$$

$$\beta_o = \frac{1}{\Omega^2}\frac{n\Omega^2\rho a c_2{}^2(B^4R_2^4 - R_1^4)}{8I_f} \tag{4.13}$$

The constants α_1, α_2 are a correction factor to compensate for simplified flybar aerodynamics, $\delta_\phi, \delta_\theta$ are the roll and pitch input command respectively, $\dot{\phi}, \dot{\theta}$ are the roll rate and pitch rate of helicopter body frame respectively, ξ is the orientation angle of flybar and $L_{5,6,7,8,9}$ are the linkages in the rotor hub assembly. After substituting the flybar flapping angle of Equation (4.12) into Equation (4.8), the equations for the moments M_ϕ, M_θ can be written in the form :

$$\begin{bmatrix} M_\phi \\ M_\theta \end{bmatrix} = \frac{n\rho\Omega^2 acR^4B^4}{16L_1(L_2+L_3)} \begin{bmatrix} \alpha_3\frac{L_3L_8}{L_9}\delta_\theta + C_1\delta_\theta - C_2\dot{\phi} + C_3\dot{\theta} - \alpha_4 L_1(L_2+L_3)\frac{\dot{\phi}}{\Omega}) \\ \alpha_3\frac{L_3L_8}{L_9}\delta_\phi + C_1\delta_\phi - C_2\dot{\theta} - C_3\dot{\phi} - \alpha_4 L_1(L_2+L_3)\frac{\dot{\theta}}{\Omega}) \end{bmatrix} \tag{4.14}$$

$$C_1 = \alpha_3 L_2 L_4 P_{f1} P_{f2}, \quad C_2 = \alpha_3 L_2 L_4 P_{f1} P_{f3}, \quad C_3 = \alpha_3 L_2 L_4 \frac{2}{\Omega} \tag{4.15}$$

$$P_{f1} = \frac{n\rho a c_2{}^2(B^4R_2^4 - R_1^4)}{8I_f}, \quad P_{f2} = \frac{\alpha_1 L_7 L_8}{L_5 L_6 L_9}, \quad P_{f3} = \frac{\alpha_2}{\Omega} \tag{4.16}$$

C_1, C_2, C_3 are terms that contributed by the flybar flapping angle Equation (4.10). The C_1 term augments the cyclic input of main rotor and C_2 term increases the damping moment in the helicopter attitude dynamics, which gives control booster to the actuator servo and stabilizing effect to the helicopter. The moment M_ψ around z-axis is defined by:

$$M_\psi = T_T L_T - K_g r + \tau_m \tag{4.17}$$

$$T_T = \frac{n}{2} a_T c_T \rho\pi R_T{}^3 \Omega_T{}^2 (\frac{B^3}{3}\theta_\psi - \frac{B^2}{2}\lambda_T) \tag{4.18}$$

where T_T is the tail rotor thrust, L_T is the length between the main and tail rotor axes, K_g is the gyro gain for tail rotor, r is the helicopter heading rate, τ_m is the main rotor induced yaw moment in the opposite direction to the tail rotor thrust, defined by:

$$\tau_m = -K_m \delta_o \tag{4.19}$$

where K_m is the main rotor torque gain, δ_o is the collective pitch cyclic input. The second term in Equation (4.17) is due to the active yaw damping system in the form of electronic gyro and is described by a simple linear model. Consequently, Equation (4.7) can then be simplified to yields the attitude dynamics:

$$\begin{aligned} \dot{p} &= k_1 qr + k_2 \delta_\theta - k_3 p + k_4 q + k_{d,1} \\ \dot{q} &= k_5 pr + k_6 \delta_\phi - k_7 q - k_8 p + k_{d,2} \\ \dot{r} &= k_9 pq + k_{10} \delta_\psi - k_{11} r - k_{12} \delta_o + k_{d,3} \end{aligned} \tag{4.20}$$

The parameters k_1 to k_{12} in Equation (4.20) are considered as the unknown parameters to be identified. The parameter $k_{d,1}$ to $k_{d,3}$ are added as constant external torque offset modelling asymmetry in roll, pitch and yaw axis.

4.1.1 Model with Dead Time

The servo actuator dynamics is an important subsystem that is directly linked to rigid body dynamics, especially implicated by the fact that the moment of helicopter is caused by the servo actuation. It is commonly modelled as a first order system:

$$\dot{\theta}_{cyc} + \tau \theta_{cyc} = k \delta_{cyc} \tag{4.21}$$

where θ_{cyc} is the blade pitch angle, τ is time constant, k is input gain constant and $\delta_{cyc} \in \delta_\phi, \delta_\theta, \delta_\psi$. In this research, the blade pitch angle or servo actuator position is considered as linear model with only input gain constant. The unmodelled time lag is accounted with additional pure time delay. The same approach is applied to the other unmodelled low order dynamics, thus the original high-order system can then be approximated with low order model with time delay which is normal practice in industry (Åström and Hägglund, 2006). Assuming there are N first-order elements in series, each having a time constant L/N , the resulting transfer function is:

$$G(s) = \frac{1}{(1+\frac{L}{N}s)^N} \tag{4.22}$$

The right hand side term is approaching pure time delay when N change from 1 to ∞, i.e.

$$e^{-Ls} = \lim_{N\to\infty} \frac{1}{(1+\frac{L}{N}s)^N} \tag{4.23}$$

Therefore, accumulation of each low time constant produces an effective time delay. Nonetheless, the dynamical effects of all the individual low order systems are mainly due to the largest time constant and can be approximated with an equivalent time constant smaller than the dominant time constant. Because of the uncertainty in the mathematical model brought about by the model approximation, we propose that model in (4.20) incorporate a dead time L so that the entire system is represented as:

$$\dot{p}(t) = k_2\delta_\theta(t-L) - k_3 p(t) + k_{d,1}$$

$$\dot{q}(t) = k_6\delta_\phi(t-L) - k_7 q(t) + k_{d,2} \tag{4.24}$$

$$\dot{r}(t) = k_{10}\delta_\psi(t-L) - k_{11} r(t) + k_{d,3}$$

where the coupling terms in (4.20) are found to have no significant effect on the overall system dynamics as will be shown in Section 5.2.1. Besides the time delay due to accumulation of time lags, the time delay is also contributed by IMU processing time and the inherent response time delay of the servo actuator. The total pure time delay L is estimated by finding the optimal correlation between the commanded angle and the measured attitude response.

The dead time e^{-Ls} is normally approximated using rational transfer function such as Pade approximation in the hope of treating an infinite-dimensional system like a finite-dimensional one. However, the problem of choosing the order of the approximation a priori is difficult, as well as proving the closed-loop stability on the basis of such a reduced open-loop model. It is concluded in (Richard, 2003) that the time delay can only be fairly approximated by rational functions at the cost of high order, which can be cumbersome for control design and even more complicated than direct synthesis from initial model with delay. The drawback of rational approximation is further aggravated by report in (Silva et al., 2001) where it is shown that the controller designed to stabilise Pade approximations may lead to unstable behaviour of the true system.

Continuous time-delay model in state space representation belongs to the class of functional differential equations (FDEs) that has infinite dimensional states. On the other

hand, when the discrete time representation is used instead, the state transition matrix dimension is greatly increased in comparison with the dead-time-free case. Hence, for control design, the transfer function model is favoured and the discrete form of system with dead time is computed as:

$$P(z) = G(z)z^{-d} \tag{4.25}$$

where $G(z)$ represents the discrete dead-time-free dynamics of the system and d is the integer multiple of sampling time that makes up the real time delay.

Dead time in the system model has great adverse effect on the closed-loop control system in terms of instability and oscillations. This can be explained through frequency domain analysis, where the phase lag introduced by the dead time decreases the phase margin (PM) of the closed-loop system and in the worst-case drives the positive PM to negative PM.

4.3 Integral-Based Parameter Identification

To identify the unknown parameters in (4.24), the integral-based parameter identification method (Hann, 2005; Hann, 2008) is significantly extended to handle high sample rates and disturbances. Specifically, the method and other published biomedical papers (Wong et al., 2006; Docherty et al., 2011) have very low sample rates. Thus, the parameters are identified over regions that have slow dynamics and very little modelling error. For the case of the helicopter, the sample rate is 1000's of times faster, so other dynamics including disturbances can occur during the time period of parameter

identification where parameters are assumed constant. Hence, the methods published in biomedical papers are not suitable for this application.

For this derivation, it is assumed that the unknown parameters in Equation (4.20) are constant for all time. The first step is to define $n+1$ time points, $T_i, \quad i = 0,...,n$ that cover the whole data range. These $n+1$ points partition the data into intervals, $\{[T_{i-1}, T_i],\ i = 1,...n\}$. For simplicity, each interval is assumed to be the same length. Integrating roll dynamics in Equation (4.20) from T_{i-1} to t yields:

$$p(t) - p_{0,i} = k_1 \int_{T_{i-1}}^{t} qr + k_2 \int_{T_{i-1}}^{t} \delta_\theta dt - k_3 \int_{T_{i-1}}^{t} p dt + k_4 \int_{T_{i-1}}^{t} q dt + k_{d,1}(t - T_{i-1}),$$
$$\mathrm{t} \in [T_{i-1}, T_i), \quad i = 1,...,n \tag{4.26}$$

Let $p_{data}(t)$ denote the measured roll rate data and define the function:

$$\begin{aligned} p_{\mathrm{model},i}(t) = p_{0,i} &+ k_1 \int_{T_{i-1}}^{t} q_{data} r_{data} dt + k_2 \int_{T_{i-1}}^{t} \delta_\theta dt \\ &- k_3 \int_{T_{i-1}}^{t} p_{data} dt + k_4 \int_{T_{i-1}}^{t} q_{data} dt + k_{d,1}(t - T_{i-1}) \end{aligned} \tag{4.27}$$

where:

$$p_{0,i} = p_{data}(T_{i-1}), \quad i = 1,..,n \tag{4.28}$$

The initial conditions in Equation (4.28) reset the beginning of each interval to the measured data, which ensures modelling error does not build up significantly over time. Choose N equally spaced time points, $\overline{T}_{data,i} = \{T_{i-1} + j\Delta t,\ j = 1,...,N\}$ in each time interval $[T_{i-1}, T_i), \quad i = 1,...,n$, with $T_0 = 0$. Setting $p_{\mathrm{model},i} = p_{data}(t),$ for $t \in \{\overline{T}_{data,i},\ i = 1,...,n\}$ gives a set of N equations in 5 unknown parameters, which is defined by the matrix equation:

$$\begin{pmatrix} I_1 \\ \vdots \\ I_n \end{pmatrix} X = \begin{pmatrix} \overline{p}_{data,1} \\ \vdots \\ \overline{p}_{data,n} \end{pmatrix} \tag{4.29}$$

where:

$$I_i = \begin{pmatrix} \int_{T_{i-1}}^{T_{i-1}+\Delta t} qr dt & \int_{T_{i-1}}^{T_{i-1}+\Delta t} \delta_\theta dt & -\int_{T_{i-1}}^{T_{i-1}+\Delta t} p dt & \int_{T_{i-1}}^{T_{i-1}+\Delta t} q dt & (T_{i-1}+\Delta t)-T_{i-1} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \int_{T_{i-1}}^{T_{i-1}+N\Delta t} qr dt & \int_{T_{i-1}}^{T_{i-1}+N\Delta t} \delta_\theta dt & -\int_{T_{i-1}}^{T_{i-1}+N\Delta t} p dt & \int_{T_{i-1}}^{T_{i-1}+N\Delta t} q dt & (T_{i-1}+N\Delta t)-T_{i-1} \end{pmatrix}, \quad i = 1,...,n \tag{4.30}$$

$$\overline{p}_{data,i} = \begin{pmatrix} p_{data}(T_{i-1}+\Delta t) - p_{0,i} \\ \vdots \\ p_{data}(T_{i-1}+N\Delta t) - p_{0,i} \end{pmatrix}, \quad p_{0,i=} p_{data}(T_{i-1}), \quad i = 1,...,n \tag{4.31}$$

$$X \in [k_1, k_2, k_3, k_4, k_{d,1}] \tag{4.32}$$

The integrals in Equation (4.30) are numerically evaluated using the trapezium rule. An approximation to the unknown parameters X in Equation (4.32) can be found by solving Equation (4.29) by linear least squares. Define the resulting parameters of this solution as $\overline{k}_1, \overline{k}_2, \overline{k}_3, \overline{k}_4$ and $\overline{k}_{d,1}$. Due to modelling error and disturbance, these resulting parameters may not be optimal after this first iteration. For the next iteration, define:

$$p_{approx}(t) = p_{approx,i}(t), \quad t \in [T_{i-1,} T_i] \tag{4.33}$$

$$\begin{aligned} p_{approx,i}(t) = p_{o,i} &+ \overline{k}_1 \int_{T_{i-1}}^{t} q_{data} r_{data} dt + \overline{k}_2 \int_{T_{i-1}}^{t} \delta_\theta dt \\ &- \overline{k}_3 \int_{T_{i-1}}^{t} p_{data} dt + \overline{k}_4 \int_{T_{i-1}}^{t} q_{data} dt + \overline{k}_{d,1}(t - T_{i-1}) \end{aligned} \tag{4.34}$$

Equation (4.33) is then substituted back into Equation (4.29)-(4.32) to form a new matrix equation. This matrix equation is solved by linear least squares to give new parameters $\bar{k}_1, \bar{k}_2, \bar{k}_3, \bar{k}_4$ and $\bar{k}_{d,1}$, which produce a second approximation $p_{approx,i}(t)$ from Equation (4.33) and (4.34). This process is continued until the least squares error between the estimated roll rate $p_{approx}(t)$ and the measured data changes less than a specified tolerance. The overall algorithm is summarized in Figure 14. Note that the partitioning of the data into n intervals is critical to account for the significant disturbance and high sampling rate in this application. Without this partitioning, the methods previously published in (Hann, 2005; Hann, 2008; Docherty et al., 2011; Wong et al., 2006) do not perform accurately on the simplified models of Equation (4.20), because of accumulated modelling error on the long stretches of data where the parameters are assumed constant.

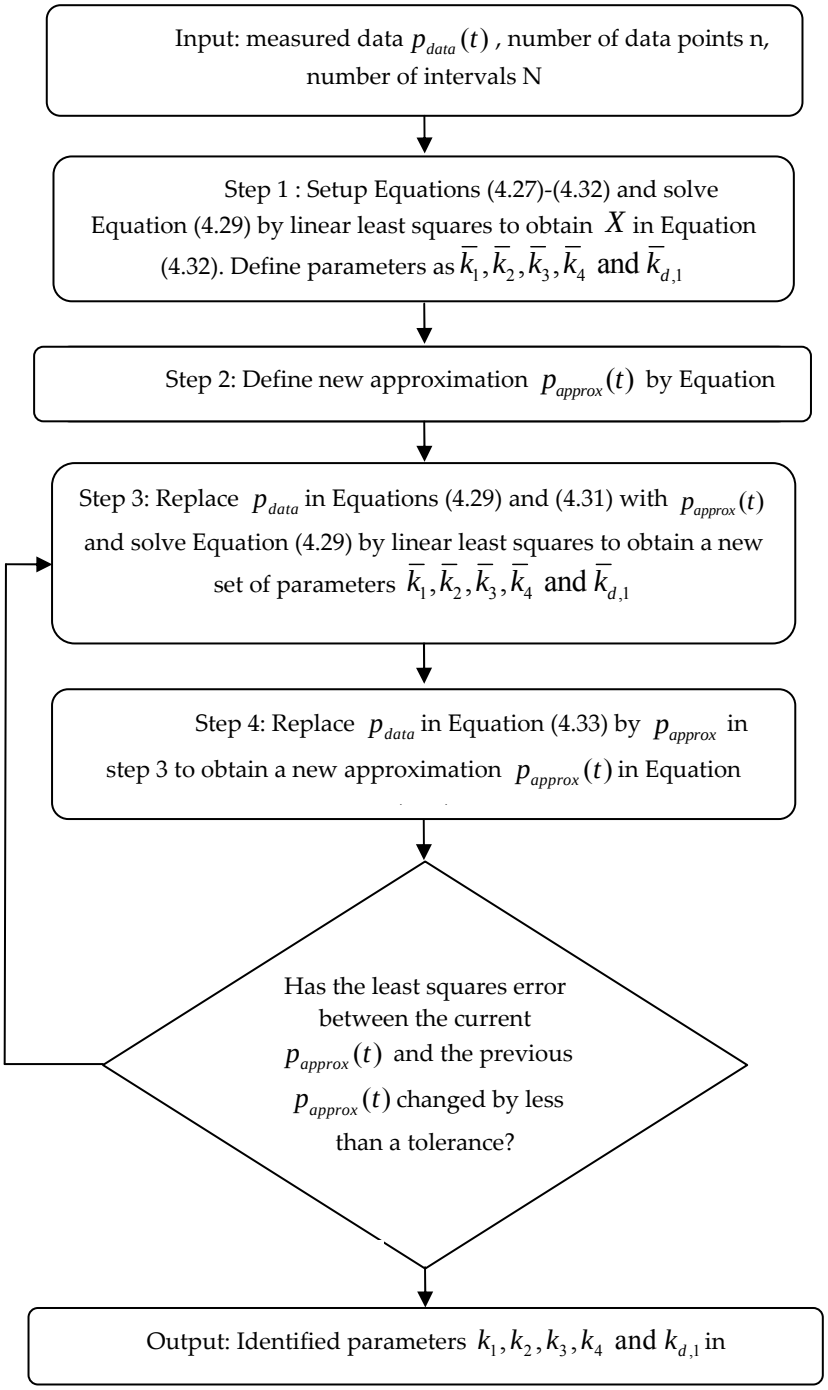


Figure 14: Algorithm of integral method.

4.3.1 Modelling Wind Disturbance

The helicopter was flown outdoors, thus environment disturbance has a significant effect on the dynamics. Wind disturbance models are common in the literature, but they typically only deal with one type of disturbance, for example vertical wind gusts (Martini et al., 2009).

The approach in this paper is to capture the complex disturbance directly by computing the effective applied torque to the helicopter at discrete intervals. The applied wind load is modelled as the constant cyclic input command required to reproduce the observed helicopter response for the given time period. The model is defined by:

$$\dot{p}(t) = k_1 q(t) r(t) + k_2 \delta_\theta(t) - k_3 p(t) + k_4 q(t) + \bar{u}_d(t) \tag{4.35}$$

$$\bar{u}_d = k_2 u_d(t) + k_{d,1} \tag{4.36}$$

In Equation (4.35), the constant values of k_1, k_2, k_3 and k_4 are assumed known or estimated and the effects of the mean wind speed, wind fluctuations and the roll offset are lumped into single parameter $\bar{u}_d$, which is defined:

$$\bar{u}_d(t) = \sum_{i=1}^{n} (H(t-(i-1)\Delta t) - H(t-i\Delta t))\bar{u}_{d,i} \tag{4.37}$$

$$\begin{aligned} H(t) &= 1, \quad t > 0 \\ &= 0, \quad t < 0 \end{aligned} \tag{4.38}$$

where $H(t)$ is the Heaviside function and Δt is the specified time interval over which $u_d(t)$ is assumed to be constant. With this formulation in Equation (4.35), the disturbance $u_d(t)$ is a meaningful parameter as it relates the external disturbance to a torque $\bar{u}_d(t)$. This torque can be written in terms of the equivalent cyclic input $u_d(t)$ and a constant disturbance

offset $k_{d,1}$ in Equation (4.36). This approach allows virtual profiles of the observed disturbances to be stored and used later for control development.

Once k_1, k_2, k_3, k_4 and $k_{d,1}$ are identified using the algorithm of Figure 14, the piecewise constant $\bar{u}_d(t)$ in Equation (4.37) is identified over the intervals of length Δt. The choice of Δt can significantly affect the model match and identified disturbance profile. If Δt is too large, the disturbance profile will not capture any high frequency wind inputs. If Δt is too small, it will capture a large amount of noise. For simplification of implementation, Δt is chosen based on the data logging frequency.

To account for noise, the identified disturbance profile is smoothed a number of times by a 10 point moving average. The number of times for smoothing is chosen to give a fitted normal distribution that is the closest least squares match to the fitted normal distribution of the blade cyclic command. The assumption is that the variations in the blade cyclic command about the mean are approximately correlated to the wind loads on the roll axis. This assumption is reasonable as the test pilot manually keeps the helicopter stable in the roll via a joy stick and radio control. If a large wind disturbance moves the helicopter in the roll, the test pilot moves the blade cyclic angle much faster and further than for the case of a small disturbance.

Let $\bar{N}$ be the number of time intervals of width Δt that fit in the whole time period of the experiment, and

$$t_i = i\Delta t, \qquad i = 1, \ldots, \bar{N} \tag{4.39}$$

Integrating Equation (4.35) from t_{i-1} to t, yields:

$$p(t_i) - p(t_{i-1}) = \int_{t_{i-1}}^{t} \left[k_1 q(t) r(t) + k_2 \delta_\theta(t) - k_3 p(t) + k_4 q(t)\right] dt + \bar{u}_{d,i}(t - t_{i-1}) \quad (4.40)$$

where $\bar{u}_{d,i}$ is an unknown constant. Solving Equation (4.40) for $\bar{u}_{d,i}$ yields:

$$\bar{u}_{d,i} = k_2 u_{d,i} + k_{d,1} \quad (4.41)$$

Once $\bar{u}_{d,i}$ in Equation (4.41) is known, $u_d(t)$ and $k_{d,1}$ in Equation (4.36) can be determined from the known value of k_2 and an assumption that $u_d(t)$ has a mean value of zero.

4.3.2 Iterative Parameter Estimation

The integral based parameter identification algorithm can be further extended to real time application where time-varying parameter model and disturbance can be estimated on the fly. This improves the prediction capabilities of a static model which could be used to improve model based control. The concept is demonstrated on the yaw dynamics but it is equally applicable to all axes. The static model of yaw model is written as:

$$\text{model}_{static} \equiv \dot{r}(t) = -k_1 r(t) + k_2(\delta_\psi(t) + k_d) \quad (4.42)$$

Equation (4.42) is a static model of the system where k_d is assumed to be constant for all time and is the baseline model for comparison. To allow for time-varying in k_1, k_2 and the external disturbance, Equation (4.24) is rewritten in the form:

$$\dot{r}(t) = k_2(t)(\delta_\psi(t) + k_{offset}(t) + ud(t)) - k_1(t) r(t) \quad (4.43)$$

The time-varying $k_d(t)$ is broken down to $k_{offset}(t)+u_d(t)$ and $k_{offset}(t)$ models the asymmetry in the helicopter which will capture any steady state offset and $u_d(t)$ models fast frequency external disturbance. The parameters k_1, k_2, k_{offset} are allowed to change over time but at much slower frequency than $u_d(t)$. Define:

$$T_{slow} \equiv \text{time period where the intrinsic parameters } k_1, k_2 \text{ and } k_{offset} \text{ are approximately constant} \tag{4.44}$$

$$T_{fast} \equiv \text{time period where the disturbance } ud(t) \text{ is approximately constant} \tag{4.45}$$

Assume that $T_{slow} > T_{fast}$ and define the time period:

$$I_{slow,t_o} = \{[t_o - T_{slow}, t_o], t_o > T_{slow}\} \tag{4.46}$$

For $t \in I_{slow,t_o}$ define the model:

$$\text{model}_{slow} \equiv \dot{r} = -k_{1,o} r + k_{2,o}(\delta_\psi + k_{offset,o}),\ t \in I_{slow,t_o} \tag{4.47}$$

where:

$$k_{1,o} \equiv k_1(t_o),\ k_{2,o} \equiv k_2(t_o),\ k_{offset,o} \equiv k_{offset}(t_o) \tag{4.48}$$

are assumed unknown parameters.

From Equation (4.44), $k_1(t) \cong k_{1,o}, k_2(t) \cong k_{2,o}, k_{offset}(t) \cong k_{offset,o}$ for $t \in I_{slow,t_o}$. Since the $k_{offset}(t)$ term in Equation (4.42) effectively ensures that the mean of the disturbance $u_d(t)$ is 0, the yaw dynamics of Equation (4.47) approximate the yaw dynamics of Equation (4.42) .

The next step is to identify the fast frequency disturbance $u_d(t)$ in Equation (4.42). For the given time t_o, define:

$$I_{fast,t_o} = \{[t_o - T_{fast}, t_o], t_o > T_{slow}\} \tag{4.49}$$

By definition of T_{fast} in Equation (4.45), $u_d(t)$ is constant for $t \in I_{fast,t_o}$ thus the yaw dynamics in this time period can be approximated by the model:

$$\text{model}_{fast} \equiv \dot{r} = -\hat{k}_{1,o} r + \hat{k}_{2,o}(\delta_\psi + \hat{k}_{offset,o} + u_{d,o}), t \in I_{fast,t_o} \tag{4.50}$$

where:

$$u_{d,o} \equiv u_d(t_o) \tag{4.51}$$

is assumed to be an unknown parameter, and $\hat{k}_{1,o}, \hat{k}_{2,o}$ and $\hat{k}_{offset,o}$ are assumed to be estimated in prior step in Equation (4.47).

In the iterative integral method for identifying the model parameters, an interval as defined in Equation (4.44) or (4.45) is used as the data range for parameters estimation instead of the whole recorded data as described in Section 4.3. This enables the algorithm to be applied in real time system where the known measurement data is only available up until current instance. First, for the intrinsic parameters identification, the parameters are found by partitioning the interval I_{slow,t_o} into n equally spaced intervals and then following the procedures as described in Figure 14.

Once the intrinsic parameters $k_{1,o}, k_{2,o}$, $\overline{k}_{offset,o}$ are identified in Equation (4.47) these parameters are substituted into Equation (4.50) and thus the only unknown parameter to be identified is $u_{d,o}$. Since this is a small interval, modelling error does not have sufficient time to build up, hence the simpler non-iterative method can be used. Equation (4.50) is first integrated from T_o to t and the steady state offset can be lumped as part of the fast disturbance, which yields:

$$\mathrm{r}(t) = r_o - \hat{k}_{1,o}\int_{T_o}^{t} r dt + \hat{k}_{2,o}\int_{T_o}^{t} \delta_\psi dt + \hat{k}_{2,o} u_{d,o}(t - T_o),\ \ t \in I_{fast,t_o} \tag{4.52}$$

Let $r_{data,\,fast}(t)$ denote the measured yaw rate data on I_{fast,t_o} and define the function:

$$\mathrm{r}_{\mathrm{mod}el,\,fast}(t) = r_o - \hat{k}_{1,o}\int_{T_o}^{t} r_{data,\,fast} dt + \hat{k}_{2,o}\int_{T_o}^{t} \delta_\psi dt + \bar{u}_{d,o}(t - T_o) \tag{4.53}$$

where:

$$\bar{u}_{d,o} = \hat{k}_{2,o} u_{d,o} \tag{4.54}$$

The parameter $\bar{u}_{d,o}$ is found by fitting $r_{\mathrm{mod}el,\,fast}(t)$ to the measured data by linear regression, and the $u_{d,o}$ can be determined from Equation (4.54) since $\hat{k}_{2,o}$ is known. In summary, the identified disturbance is defined:

$$\hat{u}_{d,o} \equiv \text{Disturbance from fitting } r_{\mathrm{mod}el,\,fast}(t) \text{ in Equation (4.53)}$$
$$\text{to the measured data by linear regression} \tag{4.55}$$

4.3.3 Model Prediction

The three models expressed in Equations (4.42), (4.47) and (4.50) are now compared by predicting ahead a given time period T_N. A prediction is made at every available data point and the residual errors between the model prediction and the actual measured value are collected for further analysis. The predictions at a time point $t = t_o$ are obtained by solving Equation (4.47) analytically in MAPLE, which yields:

$$r_{so\ln}(t) = r_o e^{-\hat{k}_{1,o}(t-t_o)} + \hat{k}_{2,o}\int_{t_o}^{t} e^{-\hat{k}_{1,o}*(t-z)}(\delta_\psi(z) + \hat{k}_{offset,o} + u_{d,o})dz \tag{4.56}$$

After some simplification, the three model predictions are thus defined:

Static prediction≡

$$r_{so\ln,\mathrm{static}}(t_o + \mathrm{T}_{fast}) = (r_o - k_d)e^{-k_1(T_{fast})} + k_d + k_2\int_{t_o}^{t_o+T_{fast}} e^{-k_1*(t-z)}\delta_\psi(z)dz \tag{4.57}$$

$k_1, k_2, k_d \equiv$ Static parameters from Equation (4.42)

Slow prediction≡

$$r_{so\ln,\text{slow}}(t_o + T_{fast}) = (r_o - \hat{k}_{offset,o})e^{-\hat{k}_{1,o}(T_{fast})} + k_{offset,o} + \hat{k}_{2,o}\int_{t_o}^{t_o+T_{fast}} e^{-\hat{k}_{1,o}*(t-z)}\delta_{\psi}(z)dz \qquad (4.58)$$

$\hat{k}_{1,o}, \hat{k}_{2,o}, \hat{k}_{offset,o} \equiv$ Slow time-varying parameters which are constant over I_{slow,t_o}.

Fast prediction≡

$$r_{so\ln,\text{fast}}(t_o + T_{fast}) = (r_o - \hat{k}_{offset,o} - \hat{u}_{d,o})e^{-\hat{k}_{1,o}(T_{fast})} + \hat{k}_{offset,o} + \hat{u}_{d,o} + \hat{k}_{2,o}\int_{t_o}^{t_o+T_{fast}} e^{-\hat{k}_{1,o}*(t-z)}\delta_{\psi}(z)dz$$

(4.59)

$\hat{k}_{1,o}, \hat{k}_{2,o}, \hat{k}_{offset,o} \equiv$ Slow time varying parameters which are constant over I_{slow,t_o}

$\hat{u}_{d,o} \equiv$ Equation (4.55)

4.4 Recursive Parameter Identification

Unlike the iterative integral-based parameter identification that directly identify the parameters in differential equation of system model, recursive parameter identification method emphasizes the aspect of discrete-time model parameters identification while dealing with possible parameters variation due to large aerodynamics variation, neglected dynamics and non-linearity. The estimation scheme updates the model parameters in real time within time limit imposed by sampling period. Thus, relatively simple algorithm is highly desirable to meet the time constraint.

On-line algorithms that are used in practice have common form such that the current parameter estimate $\hat{\theta}(t)$ is computed in terms of the previous estimate $\hat{\theta}(t-1)$ as follow:

$$\hat{\theta}(t) = f(\hat{\theta}(t-1), \varphi(\mathrm{t}), t) \tag{4.60}$$

where $\varphi(t)$ is the regressor vector that consists of present and past observations of the system outputs and inputs.

It is desirable that the estimated parameter can track the parameters of a time-varying system. In order to verify that in the analysis, it is assumed that the real system is described by models that are compatible with the models used for parameter estimation. The basic of all recursive identification algorithms is the projection algorithm, which is used in the analysis of parameters convergence. Considering the discrete-time model:

$$y(t) = \varphi(t-1)^T \theta \tag{4.61}$$

where $y(t)$ denotes the system output at time t. $\varphi(t-1)$ denotes a vector that is linear or nonlinear function: $\varphi(t-1) = \{y(t-1), y(t-2)..., u(t-1), u(t-2), ...\}$. The unknown parameters θ are to be estimated by solving the optimisation problem of:

$$J = \frac{1}{2}\left\|\hat{\theta}(t) - \hat{\theta}(t-1)\right\|^2 \tag{4.62}$$

J is minimised subject to

$$y(t) = \varphi(t-1)^T \hat{\theta}(t) \tag{4.63}$$

The resulting $\hat{\theta}(t)$ can be computed from following equation:

$$\hat{\theta}(t) = \hat{\theta}(t-1) + \frac{\varphi(t-1)}{\varphi(t-1)^T \varphi(t-1)}\left[y(t) - \varphi(t-1)^T \hat{\theta}(t-1) \right] \tag{4.64}$$

A slightly modified equation of (4.64) is formed such that the division by zero is avoided by adding a constant $c > 0$ to the denominator, and additional constant $0 < a << 1$ is added as the adaptation gain. It leads to following equation:

$$\hat{\theta}(t) = \hat{\theta}(t-1) + \frac{a\varphi(t-1)}{c + \varphi(t-1)^T \varphi(t-1)} \left[y(t) - \varphi(t-1)^T \hat{\theta}(t-1) \right] \tag{4.65}$$

As analysed in (Goodwin and Sin, 1984) that $\hat{\theta}(t)$ is never further from true parameter than $\hat{\theta}(0)$, and the modelling error $e(t) = -\varphi(t-1)^T \tilde{\theta}(t-1)$ when appropriately normalised is square summable. However, the convergence of the parameter towards the true parameter is not guaranteed.

In order to avoid oscillations that may happen in the vicinity of the optimum when the adaptation gain is not low enough, and to obtain satisfactory convergence speed when the optimum is far away which require high adaptation gain, a variation adaptation gain is desired. Least square algorithm offers such flexibility (Landau et al., 2011).

The least square algorithm is based on minimisation of least square criterion of the form :

$$\min_{\hat{\theta}(t)} J(t) = \sum_{i=1}^{t} \left[y(i) - \hat{\theta}^T(t) \varphi(i-1) \right]^2 \tag{4.66}$$

The value of $\hat{\theta}(t)$ which minimises (4.66) can be obtained by solving $\frac{\delta J(t)}{\delta \hat{\theta}(t)} = 0$. The result obtained is:

$$\begin{aligned} \hat{\theta}(t) &= \left[\sum_{i=1}^{t} \varphi(i-1) \varphi^T(i-1) \right]^{-1} \sum_{i=1}^{t} y(i) \varphi(i-1) \\ &= \bar{R}(t)^{-1} \sum_{i=1}^{t} y(i) \varphi(i-1) \end{aligned} \tag{4.67}$$

The estimation algorithm of (4.67) is in batch form, but it can be transformed into recursive form by considering:

$$\hat{\theta}(t+1) = \bar{R}(t+1)^{-1} \sum_{i=1}^{t+1} y(i)\varphi(i-1) \tag{4.68}$$

$$\bar{R}(t+1) = \sum_{i=1}^{t+1} \varphi(i-1)\varphi^T(i-1) = \bar{R}(t) + \varphi(t)\varphi^T(t) \tag{4.69}$$

Equation (4.68) can be rewritten as:

$$\hat{\theta}(t+1) = \bar{R}(t+1)^{-1} \left[\sum_{i=1}^{t} y(i)\varphi(i-1) + y(t+1)\varphi(t) \right] \tag{4.70}$$

Taking into account (4.67), (4.70) can be further rewritten as:

$$\hat{\theta}(t+1) = \bar{R}(t+1)^{-1} \left[\bar{R}(t)\hat{\theta}(t) + y(t+1)\varphi(t) \right] \tag{4.71}$$

Pre-multiply (4.69) with $\hat{\theta}(t)$ and then it is used to replace $\bar{R}(t)\hat{\theta}(t)$ in (4.71) yields:

$$\hat{\theta}(t+1) = \bar{R}(t+1)^{-1} \left[\bar{R}(t+1)\hat{\theta}(t) + \varphi(t)\left(y(t+1) - \varphi^T(t)\hat{\theta}(t) \right) \right] \tag{4.72}$$

Defines a priori prediction error as:

$$\varepsilon^0(t+1) = y(t+1) - \varphi^T(t)\hat{\theta}(t) \tag{4.73}$$

Thus, the recursive form of (4.72) is given by:

$$\hat{\theta}(t+1) = \hat{\theta}(t) + \bar{R}(t+1)^{-1}\varphi(t)\varepsilon^0(t+1) \tag{4.74}$$

which has a similar form as in (4.65) except that the gain matrix $\bar{R}(t+1)^{-1}$ is now time-varying since it depends on the past and current measurements. In order to fully transform (4.74) into a recursive form, the gain matrix has to be rewritten using the matrix inversion lemma.

Lemma 4.1 Let A, B, and C be nonsingular matrices, then the following identity holds:

$$(A^{-1} + BC^{-1}B^T)^{-1} = A - AB(C + B^T AB)^{-1} B^T A \tag{4.75}$$

Choosing $B = \phi(t)$, $C = 1$ and let $P(t+1) = \bar{R}(t+1)^{-1}$, then applying (4.75) for (4.69) yields:

$$P(t+1) = P(t) - \frac{P(t)\varphi(t)\varphi^T(t)P(t)}{1+\varphi^T(t)P(t)\varphi(t)} \tag{4.76}$$

Consequently, introduce adaptation gain $K(t+1)$ to (4.74), the formulation of the recursive least square (RLS) parameter adaptation algorithm can be formulated as:

$$\hat{\theta}(t+1) = \hat{\theta}(t) + K(t+1)\varepsilon^0(t+1) \tag{4.77}$$

$$\begin{aligned} K(t+1) &= P(t+1)\varphi(t) \\ &= P(t)\varphi(t)\frac{\varepsilon^0(t+1)}{1+\varphi^T(t)P(t)\varphi(t)} \end{aligned} \tag{4.78}$$

The RSL algorithm presented till now is based on system model of (4.61). However, due to stochastic disturbance, the system model is best described by:

$$y(t) = \varphi(t-1)^T\theta + \upsilon(t) \tag{4.79}$$

The prediction error becomes

$$\varepsilon(t,\theta) = \varphi^T(t)\theta_o - \varphi^T(t)\hat{\theta} - \upsilon(t) \tag{4.80}$$

Substitutes (4.80) in the least square criterion to solve for the difference between true parameter θ_o and the estimated parameter $\hat{\theta}$ yields:

$$\hat{\theta} - \theta_0 = (\frac{1}{n}\sum_{t=1}^{n}\varphi(t)\varphi(t)^T)^{-1}(\frac{1}{n}\sum_{t=1}^{n}V(t)\varphi(t)) \tag{4.81}$$

From (4.81), it is obvious that in order for LS estimation to be consistent, that is, for $\hat{\theta}$ to converge to θ_o, it is required that:

- $E[\varphi(t)\varphi(t)^T]$ is non-singular. This depends on the input properties, i.e. persistent excitation

- $E[\varphi(t)\upsilon(t)] = 0$. Either $\{\upsilon(t)\}$ is considered as a sequence of independent random variables with zero mean values (white noise) or $\varphi(t)$ contains only $\{u(t)\}$ terms that are independent of $\{\upsilon(t)\}$.

If the requirements are not being satisfied, a bias error exists, leading to a biased estimate of θ.

4.4.1 Recursive Prediction Error Method (RPEM)

In order to assure the above conditions, the predictor structure, the observation vector (φ) and the adaptation error (ε) have to be chosen accordingly. The two main categories of recursive identification algorithms that produce unbiased estimated parameters are identification method based on the whitening of the predictor error and identification method based on decorrelation of the observation vector and the prediction error (Landau et al., 2011). In our case, the first option is adopted where the predictor takes the form that allows whitening asymptotically the prediction error. The algorithm is derived from minimisation of prediction error criterion:

$$J_N(\theta) = \frac{1}{2}\sum_{t=1}^{N} \varepsilon^2(t,\theta) \tag{4.82}$$

It is called prediction error identification method.

Since for general model structure, the criterion will not be quadratic in θ as in (4.66), the minimisation problem becomes difficult and cannot be minimised analytically. Instead, numerical minimisation routine is required, but that typically involves several iterations through data record, and thus cannot be used in recursive algorithm.

Nevertheless, an approximation to $\hat{\theta}$ can be obtained in recursive fashion similar to the derivation for Recursive Maximum Likelihood (RML) (Ljung and Söderström, 1985). First of all, the minimisation criterion ($J_t(\theta)$) can be rewritten with the Taylor expansion around $\hat{\theta}(t-1)$ to yield:

$$\begin{aligned} J_t(\theta) = J_t(\hat{\theta}(t-1)) + J_t'(\hat{\theta}(t-1))\left[\theta - \hat{\theta}(t-1)\right] \\ + \frac{1}{2}\left[\theta - \hat{\theta}(t-1)\right]^T J_t''(\hat{\theta}(t-1))\left[\theta - \hat{\theta}(t-1)\right] \\ + o\left(\left|\theta - \hat{\theta}(t-1)\right|^2\right) \end{aligned} \tag{4.83}$$

where the prime denotes differentiation with respect to θ, and $o(x)$ denotes the higher order differentiation function. Solving the minimisation criterion (4.83) with respect to θ gives:

$$\begin{aligned} \hat{\theta}(t) = \hat{\theta}(t-1) - \left[J_t''(\hat{\theta}(t-1))\right]^{-1} J_t'\left(\hat{\theta}(t-1)\right)^T \\ + o\left(\left|\theta - \hat{\theta}(t-1)\right|\right) \end{aligned} \tag{4.84}$$

Denoting $\psi(t,\theta) \equiv \left[-\frac{d}{d\theta}\varepsilon(t,\theta)\right]^T$, first derivative and second derivative of minimum criterion (4.82) become:

$$\begin{aligned} J_t'(\theta) &= -\sum_{k=1}^{t} \psi(k,\theta)\varepsilon(k,\theta) = \left[J'_{t-1}(\theta)\right]^T - \psi(t,\theta)\varepsilon(t,\theta) \\ J_t''(\theta) &= J_{t-1}''(\theta) + \psi(t,\theta)\psi^T(t,\theta) + \varepsilon''(t,\theta)\varepsilon(t,\theta) \end{aligned} \tag{4.85}$$

With the assumption that the estimate $\hat{\theta}(t)$ is to be found in a small neighbourhood of $\hat{\theta}(t-1)$, following approximations (Ljung and Söderström, 1985) can be made:

$$\text{Neglect } o\left(\left|\theta - \hat{\theta}(t-1)\right|\right) \text{ in (4.84)} \tag{4.86}$$

$$J_t''(\hat{\theta}(t)) = J_t''(\hat{\theta}(t-1)) \tag{4.87}$$

where $\hat{\theta}(t-1)$ is the optimal estimate at time $t-1$ which leads to:

$$J'_{t-1}(\hat{\theta}(t-1)) = 0 \tag{4.88}$$

Lastly, the last term of $J_t''(\theta)$ in (4.85) is set to zero by assuming that $\varepsilon(t,\theta)$ is independent of $\varepsilon''(t,\theta)$ when close to true value θ_o. The approximation equations of (4.85) that are obtained by using the expressions of (4.86)-(4.88) are then substituted into (4.84) to yield:

$$\hat{\theta}(t) = \hat{\theta}(t-1) + \bar{R}^{-1}(t)\psi\left(t,\hat{\theta}(t-1)\right)\varepsilon^o\left(t,\hat{\theta}(t-1)\right) \tag{4.89}$$

where $\bar{R}(t)$ is the approximation of second derivative matrix $J''(\hat{\theta}(t-1))$ which is obtained by substituting (4.87) into (4.85) :

$$\bar{R}(t) = \bar{R}(t-1) + \psi\left(t,\hat{\theta}(t-1)\right)\psi^T\left(t,\hat{\theta}(t-1)\right) \tag{4.90}$$

It can be seen that (4.89) and (4.90) closely resembles RLS of (4.74) and (4.69) respectively. In fact, RLS is a special case of RPEM, which will be shown next section when computing $\psi\left(t,\hat{\theta}(t-1)\right)$. The derivation of (4.82)-(4.90) are based on the general properties and relationships between $\varepsilon\left(t,\hat{\theta}(t-1)\right)$ and $\psi\left(t,\hat{\theta}(t-1)\right)$, thus the derivation can be applied to a variety of models. On the other hand, the computation of $\varepsilon\left(t,\hat{\theta}(t-1)\right)$ and $\psi\left(t,\hat{\theta}(t-1)\right)$ depends mainly on the model used which will be discussed in next section specifically for ARMAX model.

4.4.2 RPEM Applied to ARMAX Model

The helicopter dynamics identified (4.24) can be rewritten in the form of discrete simple linear regression model:

$$A(z^{-1})y(t) = B(z^{-1})u(t) + v(t) \tag{4.91}$$

where $v(t)$ represent some disturbance in the model. Considering that in practical situations the disturbance is not just a white noise, an attempt is made to whiten the disturbance by adopting an auto regression moving average exogenous input (ARMAX) model. The ARMAX model is in the form of:

$$A(z^{-1})y(t) = B(z^{-1})u(t) + C(z^{-1})e(t) \tag{4.92}$$

Since $\{e(t)\}$ is supposed to be white noise, the term $e(t)$ is not predictable from data up to time $t-1$. Hence the best one step-ahead predictor for (4.92) is given by:

$$\hat{y}(t \mid \theta) = \left[1 - \frac{A(z^{-1})}{C(z^{-1})}\right] y(t) + \frac{B(z^{-1})}{C(z^{-1})} u(t) \tag{4.93}$$

From (4.93), the prediction error is found to be:

$$\varepsilon(t,\theta) = y(t) - \hat{y}(t \mid \theta) \tag{4.94}$$

which can be rewritten as:

$$\varepsilon(t,\theta) = \frac{1}{C(z^{-1})}\left[A(z^{-1})y(t) - B(z^{-1})u(t)\right] \tag{4.95}$$

Equation (4.95) can be put in form of:

$$\varepsilon(t,\theta) = y(t) - \theta^T \varphi(t,\theta) \tag{4.96}$$

$$\varphi(t,\theta) = \left(-y(t-1)\ \ldots\ -y(t-n_a)\ u(t-1)\ \ldots\ u(t-n_b)\ \varepsilon(t-1)\ \ldots\ \varepsilon(t-n_c)\right) \tag{4.97}$$

where n_a, n_b and n_c are the degree of polynomial $A(z^{-1}), B(z^{-1})$ and $C(z^{-1})$ respectively. In order to apply the recursive prediction error method, the gradient of prediction error $\psi(t,\theta)$ has to be computed. Equation (4.95) is derived with respect to polynomial coefficients $a_i, i = 1 \ldots n_a;\ b_j, j = 1 \ldots n_b;\ c_k, k = 1 \ldots n_c$ to yield:

$$\frac{\partial}{\partial a_i}\hat{y}(t \mid \theta) = -\frac{1}{C(z^{-1})}y(t-i) \tag{4.98}$$

$$\frac{\partial}{\partial b_i}\hat{y}(t \mid \theta) = \frac{1}{C(z^{-1})}u(t-i) \tag{4.99}$$

$$\frac{\partial}{\partial c_i}\hat{y}(t \mid \theta) = \frac{1}{C(z^{-1})}\varepsilon(t-i) \tag{4.100}$$

From the gradient computed in (4.98)-(4.100), it can be seen that the gradient vector, $\psi(t,\theta) = \left[\frac{\partial}{\partial a_i}\hat{y}(t \mid \theta), \frac{\partial}{\partial b_i}\hat{y}(t \mid \theta), \frac{\partial}{\partial c_i}\hat{y}(t \mid \theta)\right]$ is analogous to the observation vector $\varphi(t,\theta)$ being filtered by $\frac{1}{C(z^{-1})}$. Notice that when $C(z^{-1}) = 1$, the RPEM algorithm becomes RLS, which shows that RLS is a special case of RPEM. It should be noted that the residual error, $\varepsilon(t)$ is obtained by substituting $\hat{\theta}(t)$ from (4.89) into (4.96). The resulting $\varepsilon(t)$ is in turn used to construct $\varphi(t+1)$ and $\psi(t+1)$ in (4.97)-(4.100). $\varepsilon(t)$ used here is also known as posteriori prediction error as it is based on the updated $\hat{\theta}(t)$ instead and can be obtained from:

$$\varepsilon(t+1) = \frac{\varepsilon^o(t+1)}{1+\varphi^T(t)P(t)\varphi(t)} \tag{4.101}$$

where $\varepsilon^o(t+1)$ is the priori prediction error as has shown in (4.73).

The extra effort of making the use of residuals instead of prediction error is worthwhile as this results in more rapid transient convergence and greater accuracy with only slightly more computation. This is especially significant for parameters that enter non-linearly into the prediction error (Ljung and Söderström, 1985).

4.4.3 Time-Varying Parameter Estimation

In adaptive control, parameters in the model are assumed to be slowly time varying so that changing physical system dynamics can be tracked more accurately. However, the $\bar{R}(t)$ matrix in (4.90) is in integrator form that keeps on increasing with sampling data. Thus the adaptation gain, $R(t)^{-1}\psi(t)$ trends towards zero and the algorithm loses the ability to track time-varying parameters. Nonetheless, this can be handled by weighted criterion by assigning less weight to the older measurements that no longer represent the system, shown as:

$$J(\theta,t) = \mu(t)\sum_{i=1}^{t}\gamma(t,i)\varepsilon^2 \tag{4.102}$$

where $\gamma(t,i)$ increase in *i* for given *t* and has a structure such as:

$$\gamma(t,i) = \lambda(t)\gamma(t-1,i), \quad 1 \le i \le t-1 \tag{4.103}$$

which can be further elaborated in the form of:

$$\gamma(t,i) = \left[\prod_{j=i+1}^{t}\lambda(j)\right]\rho_i, \text{ where } \gamma(i,i) = \rho_i \tag{4.104}$$

$\lambda(j)$ is a parameter such that $0 < \lambda(j) \le 1$. This parameter is known as discounting or forgetting factor. It is a data weighting where most recent data is given unit weight, but

data that is n time units old is weighted by λ^n. With expression (4.103), the first derivative of criterion function (4.102) with respect to θ is obtained in terms of $\lambda(t)$ as:

$$\begin{aligned} J_t'(\theta) &= -\sum_{k=1}^{t} \gamma(t,k)\psi\left(k,\theta\right)\varepsilon\left(k,\theta\right) \\ &= \lambda(t)J_{t-1}'(\theta) - \rho_t\psi\left(t,\theta\right)\varepsilon\left(t,\theta\right) \end{aligned} \tag{4.105}$$

For the prediction-error approach, the general search algorithm (4.84) is applied to the weighted quadratic prediction error criterion (4.102), and assuming that $\hat{\theta}(t-1)$ actually minimises J_{t-1}' so that

$$J_{t-1}'(\theta) = 0 \tag{4.106}$$

Equation (4.106) is substituted in (4.105) to solve for $\hat{\theta}(t)$ in (4.84):

$$\hat{\theta}(t) = \hat{\theta}(t-1) + \bar{R}(t)^{-1}\rho_t\psi(t-1)\varepsilon^0(t) \tag{4.107}$$

Introducing

$$\bar{R}(t) = \sum_{i=1}^{t} \gamma(t,i)\psi(i-1)\psi^T(i-1) \tag{4.108}$$

Following (4.103), (4.108) becomes:

$$\bar{R}(t) = \lambda(t)\bar{R}(t-1) + \rho_t\psi(t-1)\psi^T(t-1) \tag{4.109}$$

Letting $P(t) = \bar{R}(t)^{-1}$ and considering ρ_t to be unity, $\hat{\theta}(t)$ is found to be the same one as defined in (4.89), which is:

$$\hat{\theta}(t) = \hat{\theta}(t-1) + P(t)\psi(t-1)\varepsilon^0(t) \tag{4.110}$$

On the other hand, the adaptation gain can be obtained by applying Lemma 4.1 to (4.109) to yield:

$$P(t)=\frac{1}{\lambda(t)}\left[P(t-1)-\frac{P(t-1)\psi(t\text{-}1)\psi^{T}(t\text{-}1)P(t-1)}{\lambda(t)+\psi^{T}(t\text{-}1)P(t-1)\psi(t\text{-}1)}\right] \tag{4.111}$$

The effect of the forgetting factor is mainly on $P(t)$ and hence the gain $P(t)\psi(t-1)$ in (4.110) is kept larger. $P(t)$ is kept non-zero as t approaches infinity. Thus, the algorithm is always on alert in tracking time-varying dynamics.

4.4.4 Choice of Adaptation Gain

The use of constant forgetting factor may cause a problem in adaptive regulation if $\left\{\psi(t)\psi^{T}(t)\right\}$ sequence becomes null in average. In this case, $P(t)$ in (4.111) becomes:

$$P(t)=\frac{P(t-1)}{\lambda(t)}=\frac{P(t-1)}{\lambda_{o}} \tag{4.112}$$

and

$$P(t+i)=\lambda_{o}^{-i}P(t) \tag{4.113}$$

where $P(t+i)$ and the adaptation gain grow towards infinity, which is known as estimator windup.

In our case, the variable forgetting factor is chosen, which is given by:

$$\lambda(t)=\lambda'\lambda(t-1)+1-\lambda';\quad 0<\lambda'<1 \tag{4.114}$$

$\lambda(t)$ can be viewed as the output of first order filter $(1-\lambda')/(1-\lambda' z^{-1})$. It is chosen this way to avoid the adaptation gain from decreasing too rapidly but to converge with an acceleration. In order to keep the estimator alertness to time-varying parameters, the variable forgetting factor is used with constant trace algorithm. The estimator starts with variable forgetting factor and switches to constant trace when:

$$tr(P(t)) \le nG, \quad G = 0.01 \text{ to } 4 \tag{4.115}$$

where n is the number of parameters to be estimated. In the constant trace algorithm, the forgetting factor is updated in such a way that the trace of the gain matrix is maintained at a constant value, i.e. $trP(t+1) = trP(t) = nG$. Under the given condition, the forgetting factor can be obtained by solving (4.111):

$$\lambda(t) = 1 - \frac{tr[P(t)\psi(t)\psi^T(t)P(t)]}{trP(t)\left[m^2(t) + \psi^T(t)P(t)\psi(t)\right]} \tag{4.116}$$

This combination of estimator scheme has the advantages of avoiding estimator windup in the case of no persistent excitation on the input, while providing non-vanishing adaptation gain to assure the adaptation alertness to parameter changes.

4.4.5 Robust Adaptive Parameter Estimation

It is known that in adaptive control with pure gradient-based parameter estimator, the resulting adaptive system is easily destabilized in the presence of small bounded disturbance or unmodelled dynamics (Egardt, 1979), time-varying parameters (Anderson and Johnstone, 1983) and reduced order modelling. In this section, appropriate modifications are made to the parameters adaptation algorithm presented in the previous section to circumvent the effects arising from violation of the following assumptions:

1. The true plant model and the estimated plant model have the same structure.
2. The disturbance is zero mean and stochastic in nature.

3. For parameters estimation in closed loop, the controller is able to stabilize the closed loop and has the disturbance rejection capability.
4. The parameters are constant or piecewise constant.
5. The domain of possible parameters value is not constrained in general.

For model identification in closed loop, there is a need to filter the input-output data explicitly when using the open loop estimation algorithm. When appropriate filter is chosen, the closed loop estimation will limit the frequency components in the signal to be within the band pass defined by closed loop poles of the controller (Astrom, 1993). Filtering of data signal is of crucial importance to eliminate the high frequency components which are not modelled and to focus the parameter adaptation to low frequency dynamics that is captured by the identified model. This allows low order model to be used to produce consistent parameter estimation. In addition, in plant model identification in closed loop using open loop identification algorithm, the closed prediction error is a natural result of filtering of input-output data.

However, it does not guarantee the stability of an adaptive control system in the presence of unmodelled dynamics. Additional modification known as data normalisation has to be considered. The objective is to ensure that the boundedness of adaptation error or the parameter estimated under unbounded input-output data due to the influence of unmodelled dynamics (Praly, 1983). The regressor ϕ containing inputs and outputs may grow unbounded as in the context of adaptive control, then the stability of the adaptive control scheme using lower order estimated model is not guaranteed. The normalization of

input-output data is obtained by dividing the data by a quantity $m(t)$ and the corresponding output equation has the form:

$$\bar{y}_f(t+1) = \theta^T \bar{\phi}_f(t) + \bar{w}_f(t+1) \tag{4.117}$$

with:

$$\bar{y}_f(t+1) = \frac{y_f(t+1)}{m(t)};\quad \bar{\phi}_f(t) = \frac{\phi_f(t)}{m(t)};\quad \bar{w}_f(t+1) = \frac{w_f(t+1)}{m(t)} \tag{4.118}$$

where subscript " f " denote signal filtering, $w(t+1)$ is the unmodelled response. Making the assumption that all the poles of unmodelled response transfer function are within a radius of μ ($\mu < 1$) and following rigorous derivation in (Landau et al., 2011), we arrive at normalization function as:

$$m^2(t) = \mu^2(t) m^2(t-1) + \max\left[\left\| \phi(t) \right\|^2, 1 \right];\quad \text{m(0)=1} \tag{4.119}$$

Thus far, the boundedness of estimated parameters has not been considered. Due to the integral nature of the parameter adaptation algorithm, the boundedness of the parameter estimates may be lost. Even though with the constant trace algorithm, the adaptation gain is kept within a bound, the integral nature of the estimator causes a drift in the estimation parameters. Therefore, a dead zone is introduced to the estimation algorithm (Egardt, 1979; Peterson and Narendra, 1982) such that the parameter adaptation is frozen when adaptation error is equal or below a certain level σ_o. The scheduling variable α(t) is used to adapt only when the signals have the information that can improve the parameter adaptation process. A good test for the relevance of the signals for adaptation is (Clary and Franklin, 1985):

$$\sigma(t) = \bar{\phi}_f^{\,T}(t-1)P(t-1)\bar{\phi}_f(t-1) \geq \sigma_o \tag{4.120}$$

and the scheduling variable function becomes:

$$\alpha(t) = \begin{cases} 1 & \text{if } |\sigma(t)| > \sigma_o \\ 0 & \textit{otherwise} \end{cases} \tag{4.121}$$

Taking into account the normalized variables (4.118), $\hat{\theta}(t+1)$ can be computed as:

$$\hat{\theta}(t+1) = \hat{\theta}(t) + \alpha(t)\frac{P(t)\psi(t)\varepsilon^o(t+1)}{m^2(t) + \psi^T(t)P(t)\psi(t)} \tag{4.122}$$

and

$$P(t+1) = \frac{1}{\lambda(t)}\left\{P(t) - \alpha(t)\frac{P(t)\psi(t)\psi^T(t)P(t)}{m^2(t) + \psi^T(t)P(t)\psi(t)}\right\} \tag{4.123}$$

4.5 Summary

This chapter presents the attitude dynamics of AUV helicopter in the form of decoupled linear system plus dead time. The dead time in the dynamics system introduces the phase lag to the dynamic system and affects the stability of the closed-loop control system. As the dynamic model is formed with prior knowledge of physical modelling, the number of model parameters appears to be much less compared with non-parametric modelling. This chapter also presents the integral-based parameters identification method in obtaining the unknown constant parameters in the attitude dynamics model. Furthermore, in order to account for the time-varying system parameters due to unmodelled dynamics and the external disturbances, an iterative integral-based parameter identification is proposed to estimate the time-varying model parameters and disturbance

in real time. In addition, this chapter also explores the recursive parameter identification algorithm, i.e. RPEM that can be used in the adaptive control scheme.

5. SYSTEM IDENTIFICATION RESULTS

5.1 Introduction

In order to characterise the attitude dynamics of the helicopter, outdoor flight experiment is carried out where significant external disturbance is present. The unknown model parameters are then identified using the integral-based parameter identification as well as the open-loop commands from a test pilot. In addition, a PID control experiment is also performed indoors using test rig to test the algorithm without the added wind disturbance. It provides a further validation of the methods in closed-loop as well as open-loop. The simulation of the attitude dynamics using the identified intrinsic parameters and disturbance identification shows a good match between the measured and simulation output response. Good model prediction for time period of 0.1 to 0.3 seconds are obtained using slow-varying parameters and disturbance modelling.

This chapter is organised as follows. The Section 5.2 shows the offline identification results with data obtained from outdoor flight experiment as well as the closed-loop control experiment on test rig. In Section 5.3, the iterative integral-based system identification results are shown with slow time-varying parameter identification and disturbance identification. Section 5.4 is the summary of this chapter.

5.2 Offline System Identification

5.2.1 Integral Based System Identification on Outdoor Helicopter Data

In order to collect the input-output data that would characterise the UAV helicopter behaviour in an outdoor environment, a number of manoeuvres were executed by a test

pilot via a remote control system, and the onboard avionic system recorded the necessary information for system identification. In the experiment, a varying frequency sweep input signal was used as an excitation signal to the system, in hope of revealing potential coupling in the inputs and outputs. The helicopter was initially piloted to slowly take off to a certain height, then a sinusoidal low frequency excitation signal of approximately 1~3Hz was applied on one of the cyclic inputs while maintaining a constant height. As the helicopter take-off dynamics is not considered in the model, the flight data is truncated and collected from the point when the UAV helicopter enters hovering stage. Lateral and longitudinal flight motions are performed by the pilot and the input-output data is recorded at a sampling rate of 100Hz and passed through a low pass filter which has a cut-off frequency at 15Hz to remove undesired noises such as structural vibrations. It is recommended that the filter for all the output and input is chosen with a cut-off frequency 5 times higher than the maximum frequency of the excitation signal (Tischler and Remple, 2006a).

The normalised control inputs that are converted from the recorded PWM signal are shown in Figure 15. To further show the frequency range of the input excitation, a Fast Fourier Transform (FFT) plot for each control input is given in Figure 16. The control inputs in Figures 4 and 5 include purposely constructed oscillation as well as the natural pilot response to mitigate wind gusts on the helicopter. These inputs cover a good range of excitation frequencies for identifying the major attitude dynamics in each axis. In addition, Figure 15 shows that prior to the significant actuation input from the pilot, the helicopter is held as close as possible to the steady state given the gusty wind conditions.

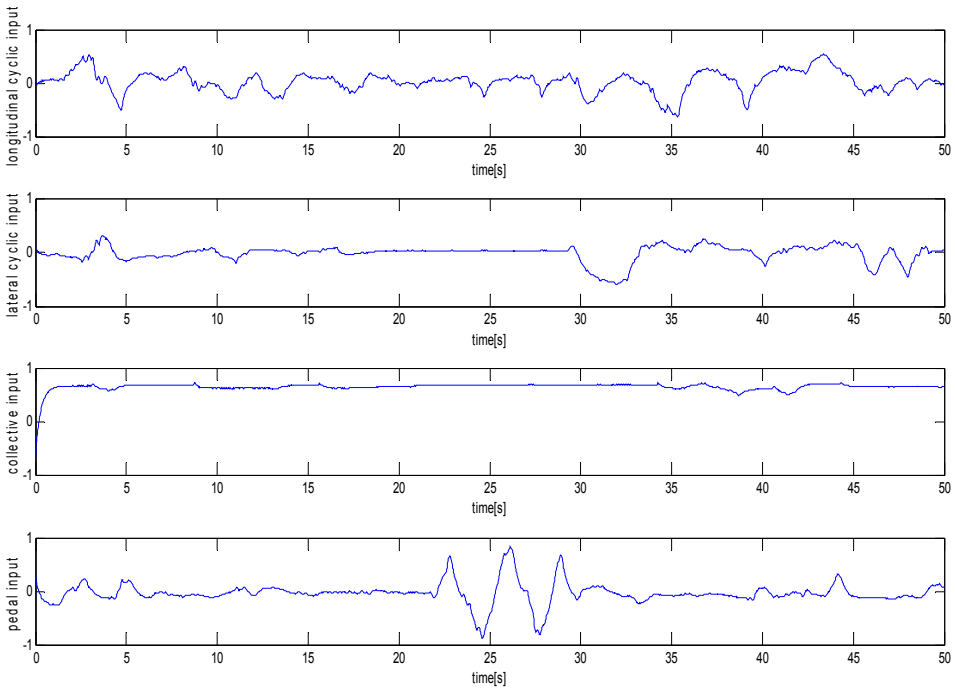


Figure 15: Control input plot.

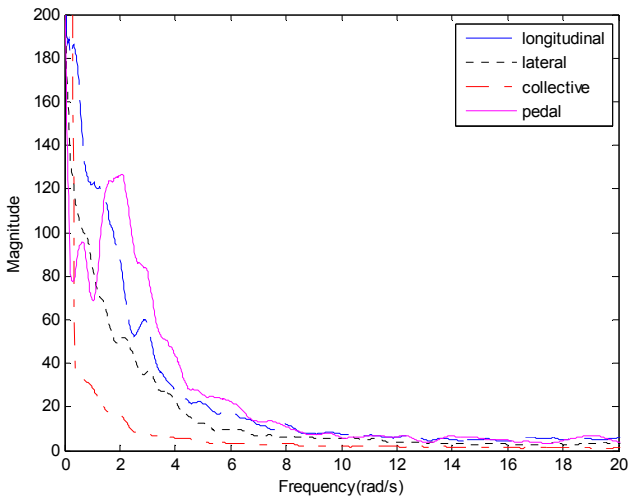


Figure 16: FFT plot of control inputs.

Figure 17(a) shows an example of the measured angular roll rate, which is quite noisy. The noise is likely due to the large amount of vibration transferred from the airframe to the sensor since there is no such noise in the control inputs of Figure 15. To reduce this

noise and allow a more suitable comparison with the model, a finite impulse response (FIR) low pass filter is implemented in MATLAB. The cut-off frequency of passband and stopband are set at $4Hz$ and $8Hz$ respectively, which is more than enough to cover the observed frequencies in the control input. Figure 17(b) gives an example of the smoothed profile.

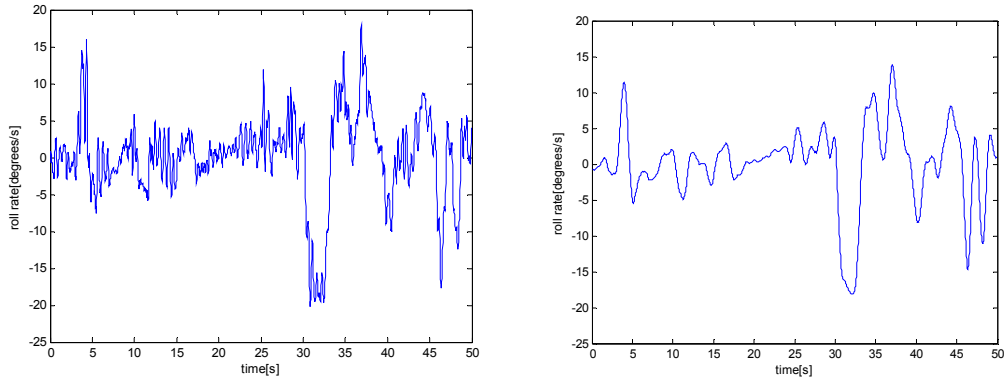


Figure 17: (a) Measured roll rate (b) Smoothed measured roll rate

The algorithm of Figure 14 is now applied to the roll rate data in Figure 17(b) with Equation (4.24). The resulting parameters are:

$$k_2 = 1.3475 rad/s^2/\delta_\phi \;, \quad k_3 = 2.4208\ s^{-1}, \quad k_{d,1} = 0.0399 rad/s^2 \tag{5.1}$$

These parameters are substituted into the model differential Equation (4.24) and numerically simulated using ode45 in MATLAB. Figure 19 gives a close up of two regions in Figure 18, which shows a good overall match. A similar algorithm is now applied to the pitch rate and yaw rate data with coupling terms set to zero, the resulting parameters are:

$$k_6 = -0.8903 s^{-1} \;, \quad k_7 = 2.5188 rad/s/\delta_\theta \;, \quad k_{d,2} = 0.0115 rad/s^2 \tag{5.2}$$

$$k_{10} = \ 13.4702 rad/s/\delta_{\psi}\ , k_{11} = 3.4342 s^{-1}\ ,$$
$$k_{12} = \ -0.5623 rad/s/\delta_{o},\ k_{d,3} = 0.0964 rad/s^{2} \tag{5.3}$$

The overlapping of the estimated pitch and yaw rate with the corresponding measured data is shown in Figure 20 and Figure 21 respectively.

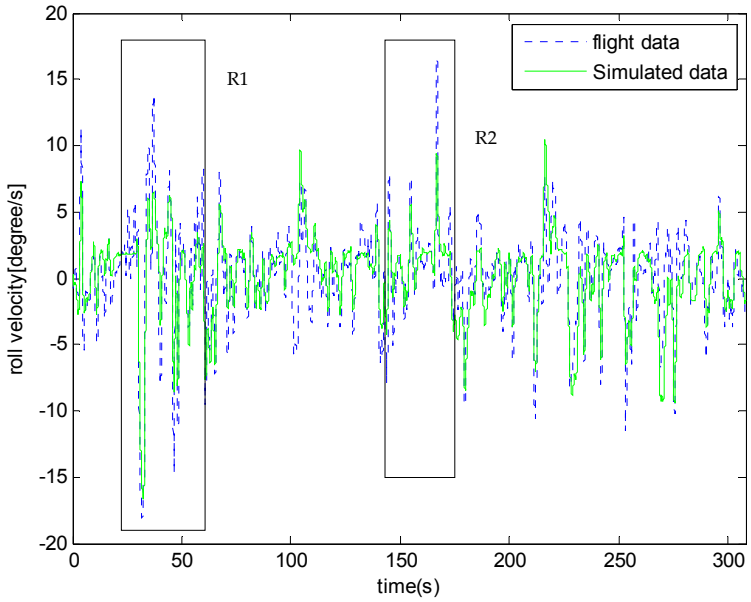


Figure 18: Roll rate estimation compared to measured data.

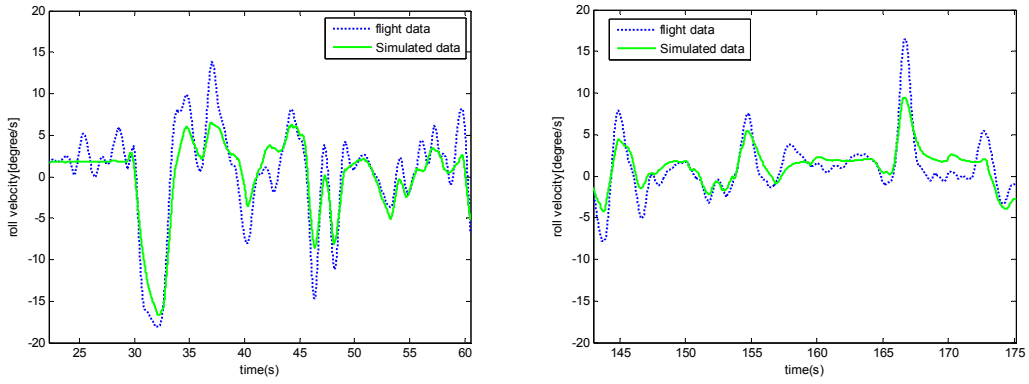


Figure 19: (a) Closed-up region R1 in Figure 8 (b) Closed-up region R2 in Figure 18.

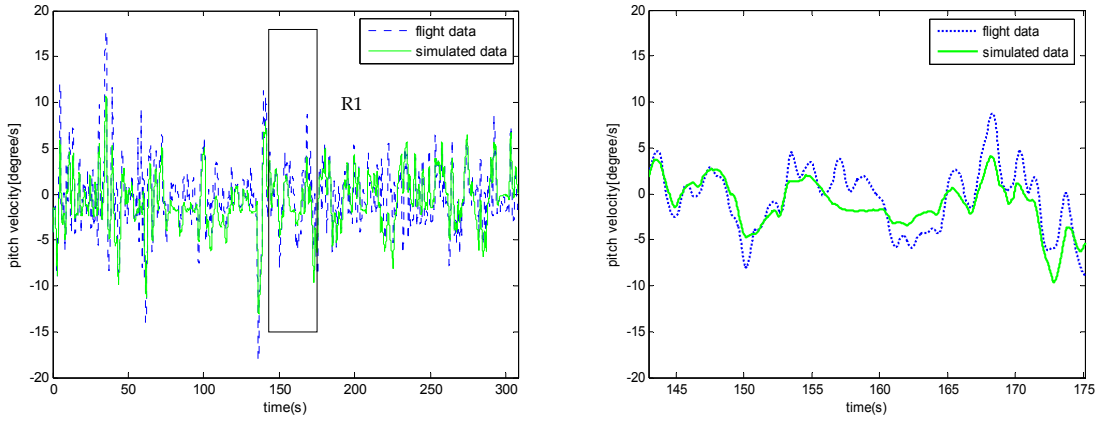


Figure 20: (a) Integral-based pitch rate estimation compared with flight data (b) Close-up region R1 in (a).

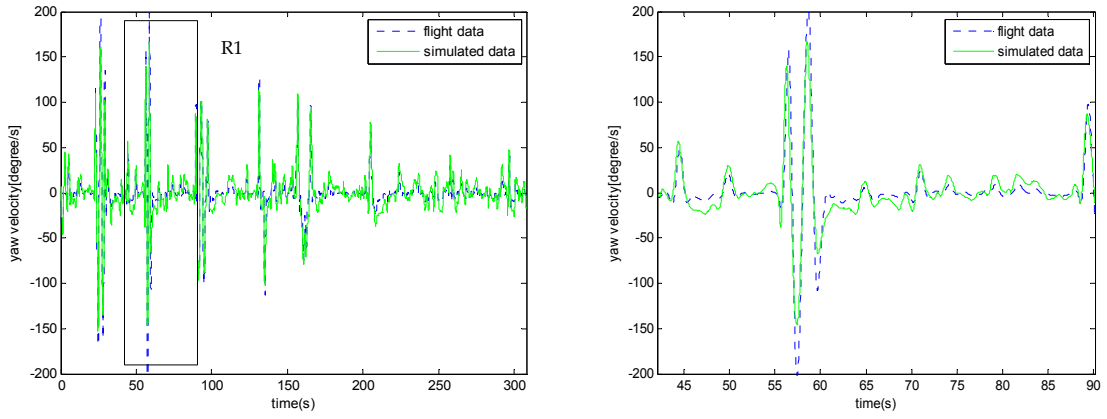


Figure 21: (a) Integral-based yaw rate estimation compared with flight data. (b) Close-up region R1 in (a).

The mean absolute error and the 90th percentile of the attitude rate in degrees/s are given in Table 1. The error of the attitude rate relative to the maximum absolute attitude rate is also shown. Note that yaw rate has significantly larger errors than the roll rate and pitch rate, but this increased error is due to the much higher yaw rate observed and it has lower relative percentage error.

Table 1: Error statistics of helicopter roll rate, pitch rate and yaw rate.

Attitude	Mean absolute error (deg/s)	90th percentile (deg/s)	Error relative to maximum (%)
Roll rate	1.5166	2.5152	8.39
Pitch rate	1.7444	3.1993	9.68
Yaw rate	7.7574	11.9200	3.83

As a further comparison, the parameters $k_1,...,k_{12}$ in Equation (4.20) are identified using the integral method. For example, in the roll rate, the parameters are identified by substituting the measured q and r and applying the algorithm in Figure 14. A similar method is applied to the pitch and yaw rates. Once the parameters are identified, Equations in (4.20) are solved numerically and the resulting response is compared with the measured data. Figure 22 gives a comparison between the coupled model response, the decoupled model response of Figure 18 and the measured data. These results further show the capabilities of the integral method to identify more complex models. It also shows that in this case there is little gain in using the complex model of Equation (4.20) over the simplified models. The full comparison is given in Table 2. The results show that the more complex model has a slightly lower mean absolute error compared with the simpler model. From the result, it is also shown that the coupling terms have very little effect on the overall attitude dynamics since any potential benefit is swamped by wind disturbance.

Table 2: Error statistics of helicopter roll rate, pitch rate and yaw rate of coupled model in addition to mean absolute error of between models.

Attitude	Mean absolute error (deg/s)		Mean absolute error(deg/s) between models	90^{th} percentile (deg/s)	Error relative to maximum (%)
	Coupled	Decoupled			
Roll rate	1.4927	1.5166	0.3669	2.4578	8.25
Pitch rate	1.6266	1.7444	0.7911	2.7739	9.03
Yaw rate	7.8203	7.7574	0.9606	12.2085	3.86

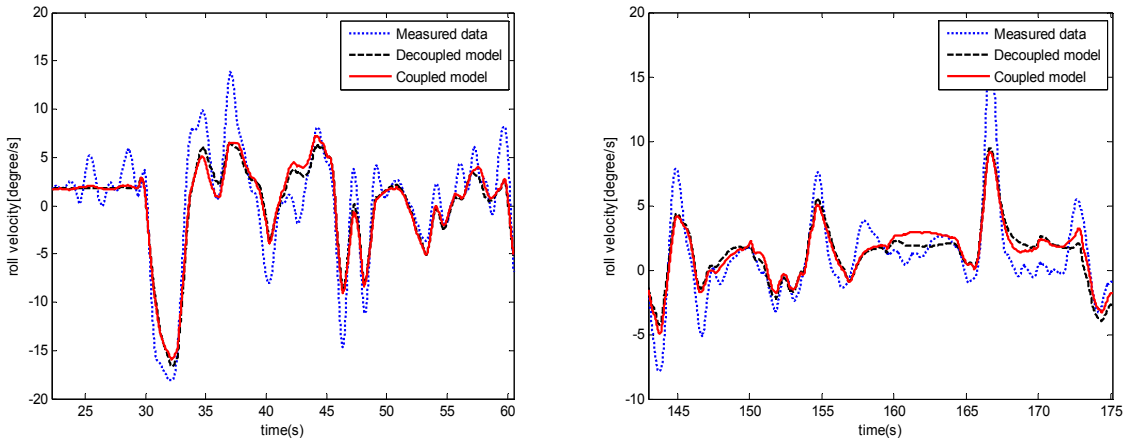


Figure 22: Comparison of roll rate between coupled and decoupled models and the measured data for (a) interval R1 (b) interval R2.

In order to further validate the dynamics model, the modelled predicted ϕ, θ and ψ of the helicopter are computed with representation in quaternion angles. The measurement of orientation angle and attitude velocity are collected using the IMU in an inertial reference frame and body fixed reference frame respectively, which are denoted as quaternion angles and angular rates. Quaternions are used to calculate the attitude angles to avoid the common problem of gimbal lock with Euler angles, that can cause significant numerical errors. The attitude angles in quaternions are computed by integrating the kinematic equation in terms of the unit quaternion vector describing the relation between the rigid body attitude variation and the body angular velocities. The initial conditions are assumed to be zero rotation angle. The quaternions vector $\begin{bmatrix} q_0 & q_1 & q_2 & q_3 \end{bmatrix}$ is solved by the standard set of differential equations:

$$\begin{bmatrix} \dot{q}_0 \\ \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{bmatrix} = \begin{bmatrix} -\frac{1}{2}(q_3 r + q_2 q + q_1 p) \\ \frac{1}{2}(q_2 r - q_3 q + q_0 p) \\ \frac{1}{2}(-q_1 r + q_0 q + q_3 p) \\ \frac{1}{2}(q_0 r + q_1 q - q_3 p) \end{bmatrix} \tag{5.4}$$

where:

$$\sqrt{q_0^2 + q_1^2 + q_2^2 + q_3^2} = 1, \quad q_0(0) = 1, q_1(0) = 0, q_2(0) = 0, q_3(0) = 0 \tag{5.5}$$

The time-varying rotational matrix corresponding to the unit quaternion $\begin{bmatrix} q_0 & q_1 & q_2 & q_3 \end{bmatrix}$ is defined as:

$$R_t = \begin{bmatrix} 2(q_0^2 + q_1^2) - 1 & 2(q_1 q_2 + q_0 q_3) & 2(q_1 q_3 - q_0 q_2) \\ 2(q_1 q_2 - q_0 q_3) & 2(q_0^2 + q_2^2) - 1 & 2(q_0 q_1 + q_2 q_3) \\ 2(q_0 q_2 + q_1 q_3) & 2(q_2 q_3 - q_0 q_1) & 2(q_0^2 + q_3^2) - 1 \end{bmatrix} \tag{5.6}$$

The orientation of the helicopter in terms of the body fixed frame is obtained using the equations:

$$\begin{aligned} RA &= \begin{bmatrix} RAx \\ RAy \\ RAz \end{bmatrix} = Rt * XA_0 \\ PA &= \begin{bmatrix} PAx \\ PAy \\ PAz \end{bmatrix} = Rt * YA_0 \\ YA &= \begin{bmatrix} YAx \\ YAy \\ YAz \end{bmatrix} = Rt * ZA_0 \end{aligned} \tag{5.7}$$

where $XA_0 = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}, YA_0 = \begin{bmatrix} 0 & 1 & 0 \end{bmatrix}, ZA_0 = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}$ are the unit vectors in earth reference frame, RA, PA, YA are the resulting rigid body axis vectors. The angle of body frame axis with respect to earth reference axis is then calculated using trigonometric functions:

$$\begin{aligned} &\phi = \sin^{-1}(r), \quad r = PAz / \cos(\theta) \\ &\theta = \sin^{-1}(RAz) \\ &\psi = \cos^{-1}(RAx / \sqrt{RAx^2 + RAy^2}) \end{aligned} \tag{5.8}$$

$$\begin{aligned} &\phi \equiv \text{roll angle about RA axis} \\ &\theta \equiv \text{pitch angle about PA axis} \\ &\psi \equiv \text{directional angle about YA axis} \end{aligned} \tag{5.9}$$

Figure 23 shows the axes and angles relative to the helicopter body axis, where:

$RA \equiv$ Axis parallel to the tail of helicopter from front nose

$PA \equiv$ Axis pointing directly to the right of helicopter (directly south if the nose is pointing East)

$YA \equiv$ Vertical axis perpendicular to main helicopter blades point down in the formation

The parameters ϕ, θ and ψ in Equation (5.9) by right hand rule are used to give an intuitive and easily visualized idea of the range of attitude dynamics of the helicopter during flight.

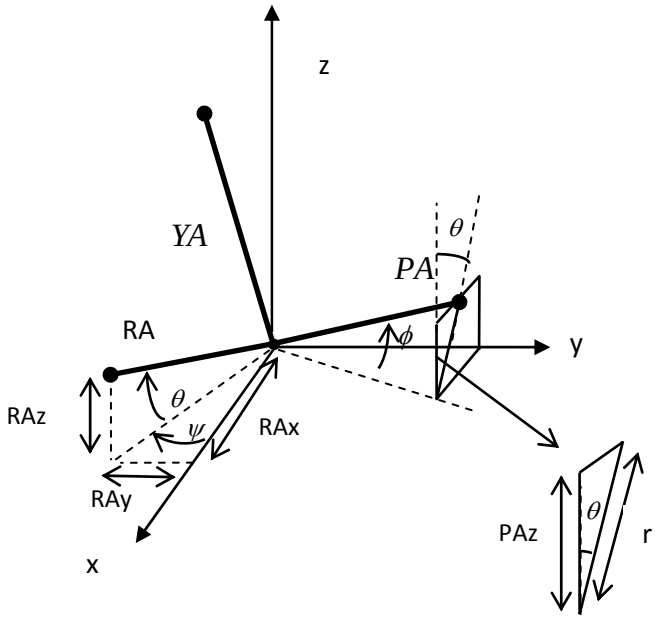


Figure 23: Angle of helicopter relative to earth reference frame.

To remove drift in integration, the best least squares piecewise linear function over 10s periods is subtracted from the data and the model using the command "detrend" in Matlab. The resulting detrended modelled and measured angles are plotted in Figure 24-Figure 26. The results show that the minimal model captures all the major attitude dynamics quite accurately.

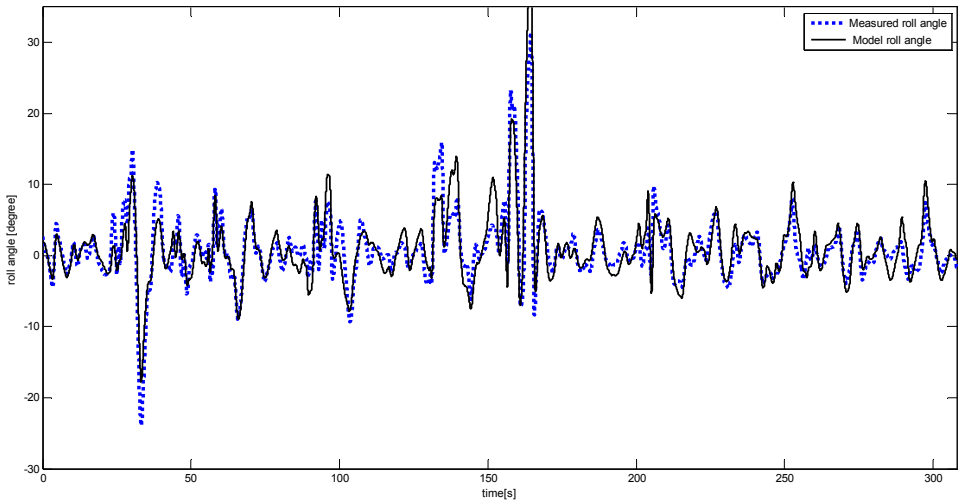


Figure 24: Measured roll angle vs quaternion derived roll angle.

Specifically, the modelled roll angle tracks the measured angle with a mean absolute error of 1.7860 degrees and a standard deviation of 2.4779 degrees. For the pitch angle, the mean absolute error is 2.1029 degrees and standard deviation is 3.0539 degrees. For the yaw angle, it has a mean absolute error of 6.8740 degrees and standard deviation of 8.9974 degrees. Note that some of the precise quantitative variations are not captured since there were significant wind gusts during the experiment and no disturbance is included in this model response. As a summary, a plot of combination of control inputs, measured and simulated angle and rate are is shown in Figure 27 to outline the interactions between the input and the response. The data interval of 20 to 35s is chosen due to the active input perturbation.

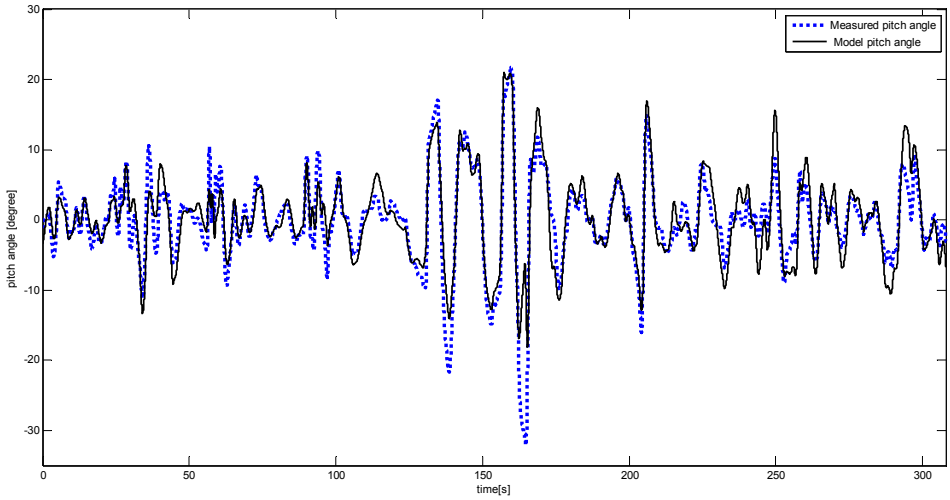


Figure 25: Measured pitch angle vs quaternion derived pitch angle.

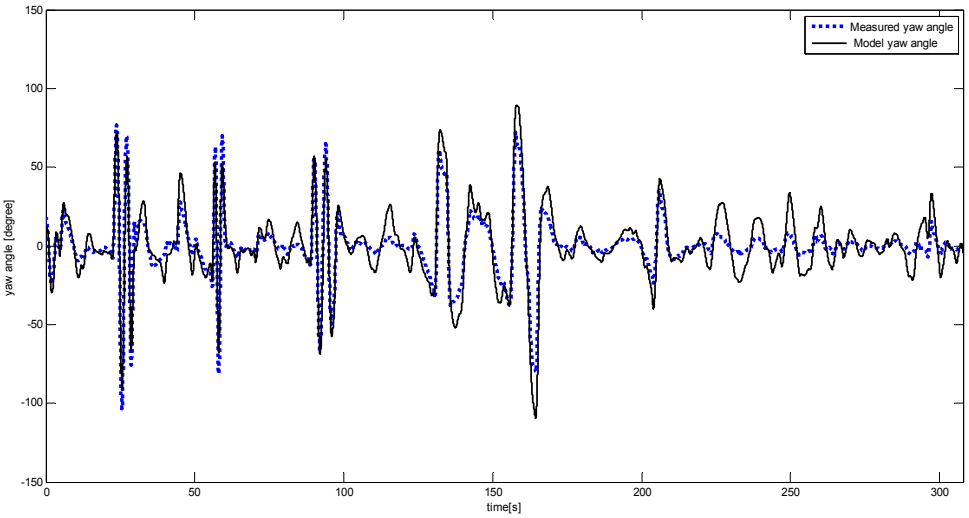


Figure 26: Measured yaw angle vs quaternion derived yaw angle.

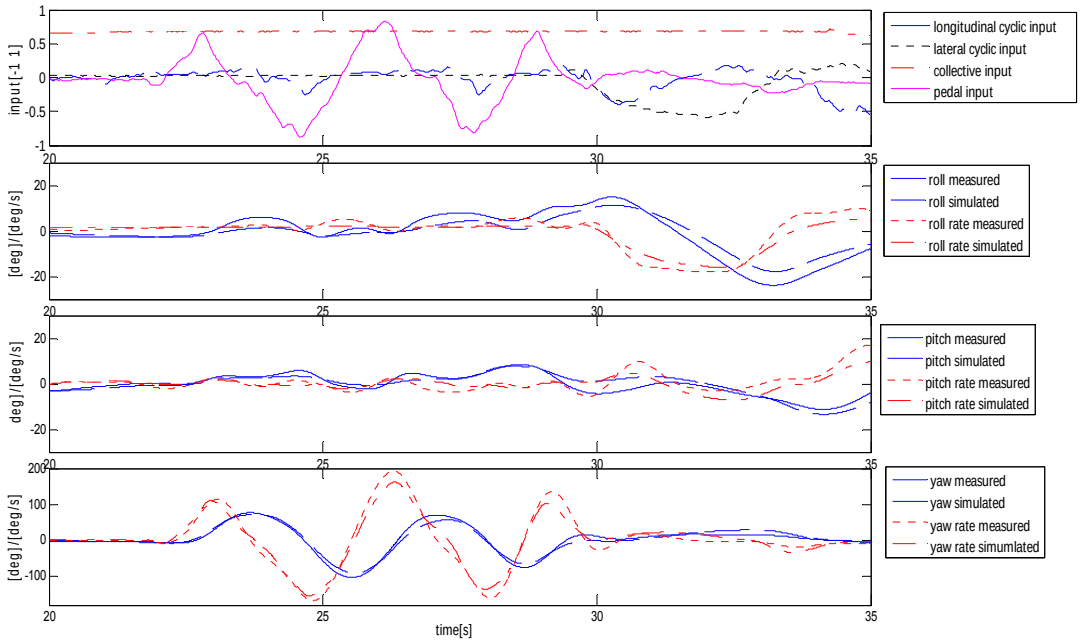


Figure 27: Combination plots of inputs, measurement and simulated angle and rate.

5.2.2 Computation Speed Evaluation

To show the significant computation efficiency of the integral method, a comparison is made to standard non-linear regression (NLR) in MATLAB. Both methods are implemented on a laptop equipped with 2.50GHz CPU and 3.98GB of RAM. In the NLR method, the objective function for Equation (4.20) is defined:

$$F(k_1, k_2, k_d) = \sum_{i=1}^{N} (p_{numerical}(k_1, k_2, k_d)(t_i) - p_{data}(t_i))^2 \tag{5.10}$$

$$p_{numerical} \equiv \text{Numerical solution of Equation (4.20)} \tag{5.11}$$

The numerical solver chosen is ode45 and the absolute and relative tolerances that control the automatic step size are increased until the error in simulation reaches a

maximum of 5% with respect to the most accurate solution obtained from using a very small step size. This approach ensures the simulation speed is made as fast as possible to provide an accurate computational comparison. The command *lsqnonlin* in MATLAB is used to find the unknown parameters k_1, k_2 and k_d that produce the best least squares solution to Equation (5.10). The starting point for the parameters is set as:

$$k_1 = 5, \quad k_2 = 5, \quad k_d = 5 \tag{5.12}$$

The results are shown in Table 3. It can be seen that integral method is approximately 710 times faster than the NLR method, and has very similar accuracy. The reason for such a significant speed increase is that the integral method only requires sums of data, which are very fast to compute and most importantly it does not require any forward simulation at each iteration. The nonlinear regression requires the solution of the underlying differential equation at each iteration, which is very computationally expensive. Similar gains have been observed in the bio-medical field (Hann, 2005).

Table 3: Comparison between Integral method and NLS.

	CPU Time(s)	**Mean Absolute error(deg/s)**
Integral Method	~ 0.63	1.5261
NLR [5,5,5]	~ 447.80	1.3913
NLR [50,50,50]	~688.00	2.4547
NLR [100, 100,100]	~ 380.06	55.7191

Table 3 also shows two cases where the starting points are set far away from the solution. One of these cases has a significant error, showing that in this case the NLR

method has found a local minima. Therefore, there is no guarantee in finding the global minima unless the starting point is near to the solution. This result shows the typical starting point dependence of NLR, which increases the computation time further when several other starting points have to be used. The integral method uses the measured data as starting point, thus it does not require a good initial starting point for the parameters.

5.2.3 System Identification and Model Prediction of Test Rig Data

To further test the algorithm of Figure 14, a PD controller on the yaw angle is implemented with a gain of K_p=0.001 and K_d=0.0001. Note that an infinite impulse response (IIR) low pass filter is implemented on the yaw angle data to reduce the effect of measurement noise. One source of the measurement noise is from the vibration of the helicopter frame, which is picked up by the accelerometer in the IMU. This vibration is only translational so should have minimal effect on the yaw angle. However, the accelerometer is part of the data fusion in the sensor's Kalman filter to reduce the drift so the weighting from this vibration causes unwanted noise in the yaw angle. Figure 28 shows a step response for a PD controller based on taking direct data from the sensor, which has noticeable oscillation. To reduce this sensor noise an IIR low pass filter is implemented with 15*Hz* cutoff frequency. Figure 29 shows another step response with the same gain after applying the filter, which shows a considerable improvement.

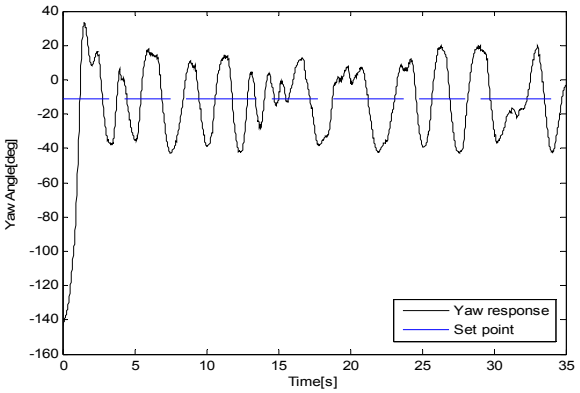


Figure 28: Yaw response without filtering.

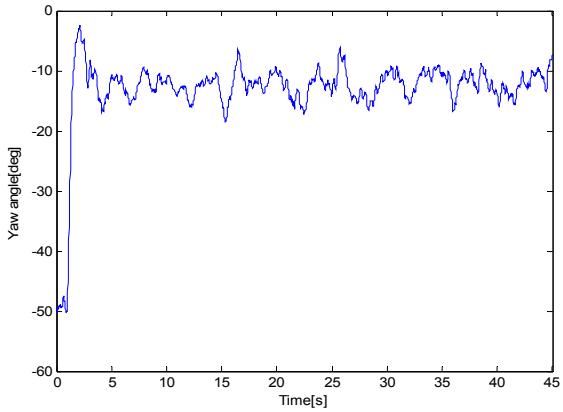


Figure 29: PD control (Kp=0.01, Kd=0.001) yaw response with filtering.

Note that an alternative solution to the filter which gives a much improved result would be to buy a much more expensive sensor, for example a laser gyro. However, for a basic proof of concept the low-cost IMU is adequate and has the advantage that it can be readily replaced.

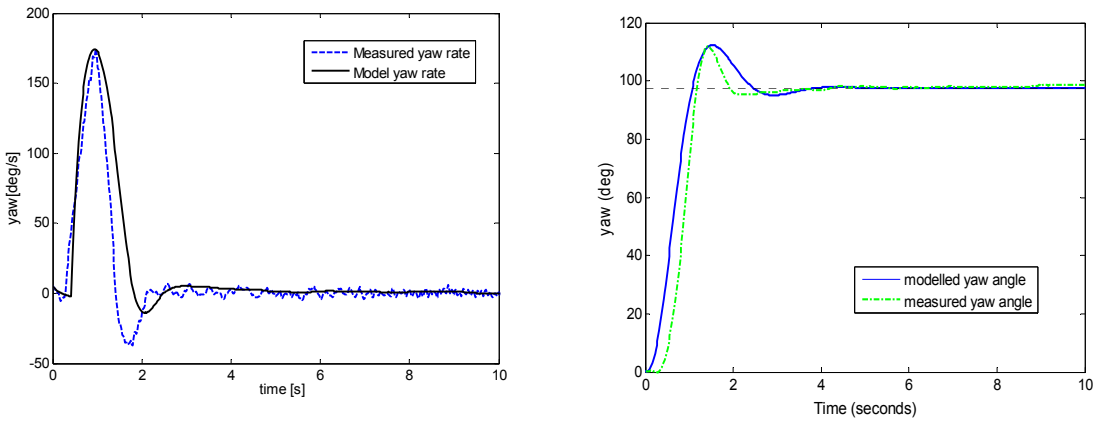
The underlying model for a PD controller is defined as:

$$\dot{r} = -k_1 r + k_2(\delta_\psi + k_d) \tag{5.13}$$

$$\delta_\psi(t) = K_p(r_o - \int_0^t r) + K_d(-r) \tag{5.14}$$

$r_o \equiv$ reference yaw angle, $K_p, K_d \equiv$ proportional(P) and derivative(D) gains (5.15)

A similar method to Figure 14 is used to identify the parameters k_1, k_2 and k_d, which are assumed constant throughout, since there is no wind disturbance on the test rig indoors. The only difference is that δ_ψ in Equation (4.53) is no longer fixed throughout all the iterations but updated in step 4 of Figure 14 by substituting $r_{approx}(t)$ into the r value in Equation (5.14) which then goes into Equation (4.53). Figure 30 shows the result of the yaw and yaw rate.



(a) (b)

Figure 30: (a) Modelled vs experiment yaw rate for PD control on test rig (b) Modelled vs experiment yaw angle for PD control on test rig.

The time delay is evident from the experiment result and is approximated in the form of first order lag rational function. The approximated time delay is then simulated along with the baseline model and PID control in closed loop. From Figure 30, it can be seen that the dominant trend in yaw dynamics are captured. The remaining error that this model cannot capture could be due to the inexact knowledge and error from the

approximation of system time delay. To account for this the model predictive method of Equation (4.50) is used with disturbance interval T_{fast}=0.1s and the yaw angle predicting length is 0.2 seconds. The new curve is given in Figure 31 for the first 10 seconds. Both the yaw rate and yaw angle are predicted very accurately with this method, demonstrating the

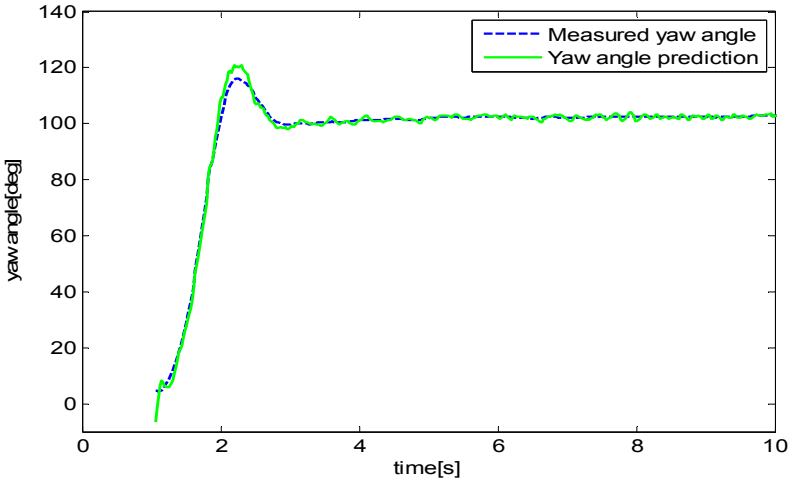
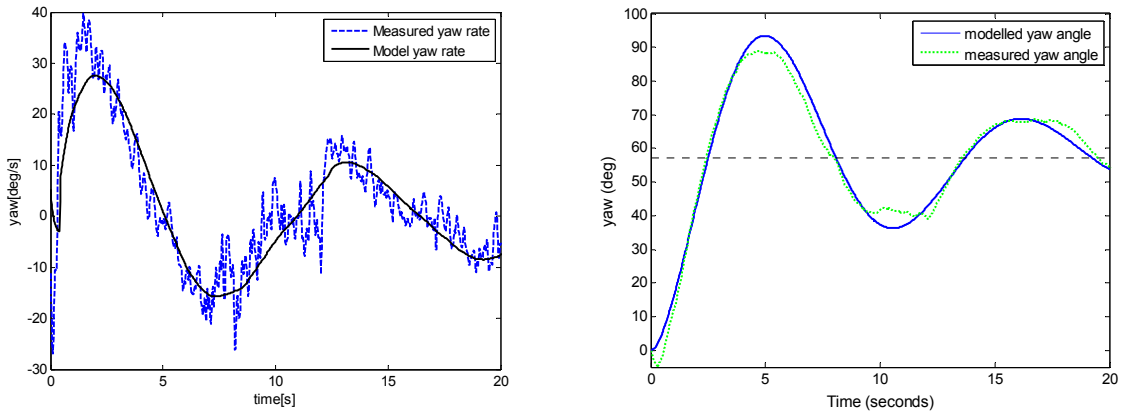


Figure 31: PD yaw angle with disturbance prediction.

potential for using this model in real real-time to improve the control response.

A further experiment is now performed with K_p=0.001, K_d=0.0001 and K_I=0.0001. The same parameters of the system static model that are used in the first experiment are retested with the second experiment control input to predict the yaw response. Figure 32 shows the results. The results show very good prediction of the overall yaw dynamics in this PID controlled response. Finally, using the identified parameters of the first experiment, the fast model prediction method with disturbance interval T_{fast}=0.8s is used to predict the yaw angle. The modelled yaw angle is the result of predicting forward 0.4

seconds at each measured time point. The resulting prediction versus measured data for the first 10 seconds is given in Figure 33. Very accurate predictions are obtained, thus showing this simplified modelling and disturbance identification are very effective in capturing the non-linear helicopter dynamics in real-time.



(a) (b)

Figure 32: (a) Modelled vs experiment yaw rate for PID control on test rig (b) Modelled vs experiment yaw angle for PID control on test rig.

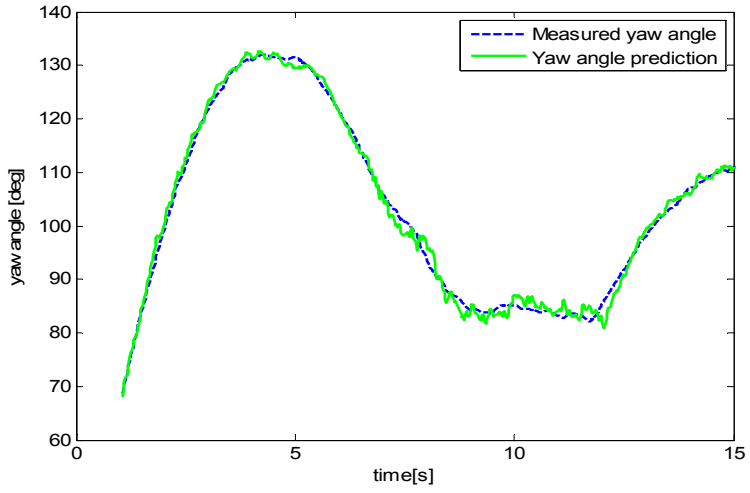


Figure 33: PID yaw angle with disturbance prediction.

5.3 Iterative Parameter Identification

The iterative integral method is now applied to identify the potential time-varying intrinsic parameter in a baseline static model, specifically in manual piloted outdoor flight experiment where low sinusoidal excitations were performed on the yaw axis in the presence of significant wind disturbance. The parameters in the baseline static model is first identified with the algorithm of Figure 14, then the comparison with the model prediction approaches of Equations (4.47) and (4.50) are made to characterise the effectiveness in improving the model prediction.

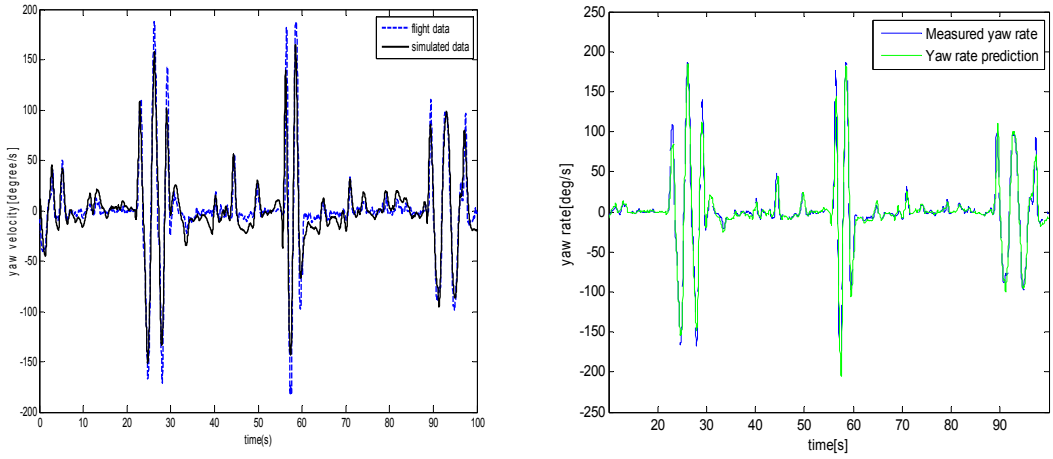
Firstly, to investigate the effect of the time period T_{slow} in Equation (4.47), several values of 5,10, 15*s* are chosen with a prediction interval of 0.1*s*. For example with T_{slow}=5*s*, starting at t_o=5s, the algorithm of Figure 14 is applied to identify $k_{1,o}, k_{2,o}$ and $\bar{k}_{offset,o}$. Then these values are used to predict the future yaw at t=5.1s using the analytical formula of Equation (4.58). The sampling period is 0.02s, and the calculation is repeated for t=5.02, 5.04, 5.06, ... The residuals of all the predictions are stored and the results are given in Table 4 including a comparison with the predictions from the static model using the parameters of Equation (5.3) and Equation (4.57). To further see these results visually, Figure 34 overlays the predicted values with the measured values for both the static and slow model cases.

The result shows that the major dynamics are captured and the discrepancies are due to unmodelled dynamics which can be further improved with time varying parameter identification. The result from Table 4 shows that T_{slow}=10s is the optimal value for lowest model error. For the T_{slow}=15s, the model error is higher compared to 10s which indicates

that this interval is not sufficiently short to capture some of the fast changing disturbance acting on the helicopter. The yaw rate error for all three values of T_{slow} is significantly lower than the yaw rate error for the static model, as expected since the static model cannot react to changes in the attitude dynamics. This significant improvement demonstrates that the wind has a significant effect on the attitude parameters over time.

Table 4: Predictions of the slow model for several values of T_{slow}.

T_{slow}(s)	Yaw rate error (slow model)		Yaw rate error (static model)	
	Median (deg/s)	90% CI (deg/s)	Median (deg/s)	90% CI (deg/s)
5	0.1018	[-5.7211 7.6100]		
10	0.0155	[-5.3370 6.8334]	0.1147	[-12.12 11.77]
15	0.2082	[-6.1265 7.4097]		



(a) (b)

Figure 34: (a) Prediction results for static model (b) Prediction results for slow model.

The effect of the disturbance identification and prediction with Equations (4.52)-(4.55) and Equation (4.59) is now investigated. Values of T_{slow}=10s and T_{fast}=0.3s are chosen. Initially t_o=10s and a moving window of 10s is shifted across all the data in a similar way to Table 4, but with 3 different prediction intervals of T_N =0.1, 0.2, 0.3s. The results are given in Table 5 and the predicted versus measured values are plotted in Figure 35.

Table 5: Predictions of the fast model for several values of prediction intervals.

	Yaw rate error (deg/s)	
Prediction intervals	**Median (deg/s)**	**90% CI (deg/s)**
0.1	0.0501	[-4.0006 4.9103]
0.2	0.0635	[-5.0578 6.3018]
0.3	0.1278	[-6.1279 7.5973]

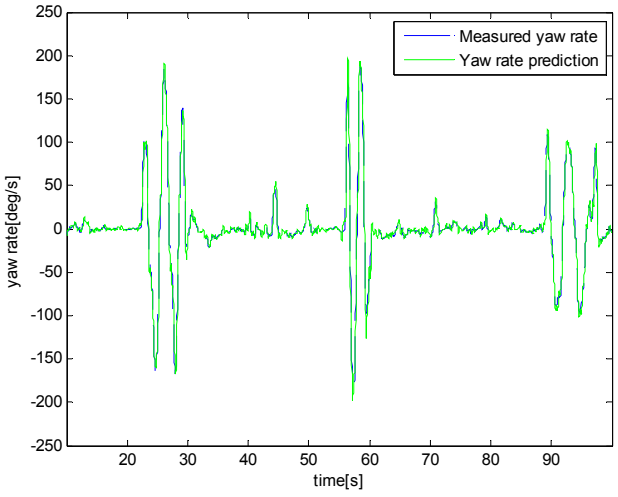


Figure 35: Prediction results for fast model.

It can be seen that the 90% CI is significantly reduced with predicting forward 0.1s and the errors are comparable to Table 4 even for the larger prediction times of 0.2 and 0.3s. This ability to predict forward accurately allows model-based control to be deployable, which will be shown in section 6.5.

As a further comparison, the static model parameters from Equation (5.3) are used instead of updating the $k_{1,o}, k_{2,o}$ and $\bar{k}_{offset,o}$. Disturbance is identified and the yaw rate is predicted forward for T_N =0.1,0.2 and 0.3s in a similar way to Table 5. The results are given in Table 6.

Table 6: Prediction of fast model based on static model.

	Yaw rate error (deg/s)	
Prediction intervals	**Median (deg/s)**	**90% CI (deg/s)**
0.1	0.0501	[-4.0006 4.9103]
0.2	0.0635	[-5.0578 6.3018]
0.3	0.1278	[-6.1279 7.5973]

Interestingly, the results of Table 6 are quite similar to Table 5. The reason is that the slow time-varying parameters are lumped into this fast time-varying disturbance. However, the advantage of the method of Table 5 is that no prior knowledge of the helicopter is required, while in Table 6 a static model must be available.

5.5 Summary

The results of system identification are validated against the real characteristic of helicopter attitude through the error statistics (i.e. mean absolute error, 90th percentile and error relative to maximum) between the simulation and the measured output response. The system identification results obtained from both open-loop and closed-loop experiments show a good match between the simulation and the measured response, which suggests that the model is able to capture the dominant dynamics of the helicopter. In terms of computation time, the integral method was 15,667 times faster than non-linear regression. The integral method also has a similar accuracy to non-linear regression method.

A further model validation is carried out with model prediction error median and 90% confidence interval. The results show that the slow model identification of helicopter's intrinsic parameters with interval of 10s combined with a fast identification interval of 0.3s to capture disturbance yields the best results. However, the combination of a static model and a fast identification interval of 0.3s give slightly worse but reasonably comparable results. This result suggests that with offline identification to create an initial static helicopter model, a real-time disturbance identification algorithm would be sufficient to account for modelling and wind disturbance in flight. This option would be much easier to implement on a chip, as the iterative integral method only requires integrals of the data and linear least squares.

6. MODEL PREDICTIVE CONTROL (MPC) FOR AUV HELICOPTER ATTITUDE CONTROL

6.1 Introduction

Two time scale approach as shown in Figure 36 is adopted in designing the overall control system for the UAV. The control system is broken into attitude control and position control. The attitude controller is considered as the inner loop control and the position controller as the outer loop control. According to the given reference position $[x_d, y_d, z_d]$, the position controller outputs the corresponding desired attitude $[\phi_d, \theta_d, \psi_d]$ of the fuselage. That value, in turn, acts as a reference input value to the attitude controller, where an appropriate cyclic pitch control $[\delta_{lat}, \delta_{lon}, T_t]$ is produced. Nevertheless, this research mainly focuses on the inner loop control, which can be challenging as fast controller response is needed.

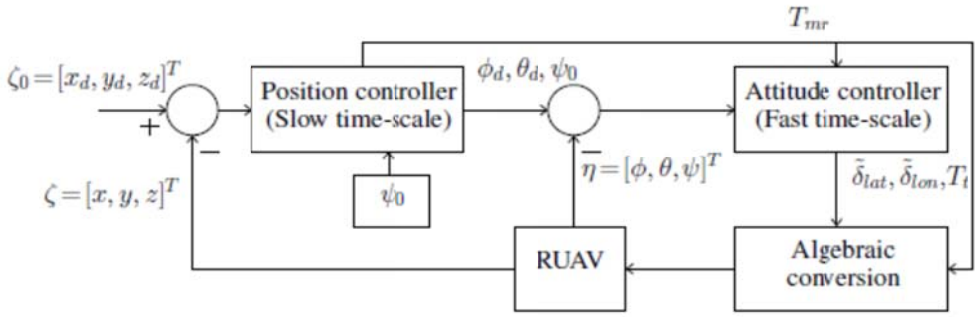


Figure 36: Two-time scale based control of RUAV (Ahmed et al., 2010).

Model predictive control (MPC) is chosen for the design of inner loop control because of its advantages over other control methods, which will be detailed later. As the process model plays a decisive role in the controller, it serves as a benchmark as to how

accurately the model identified (Chapter 4) describes the actual system dynamics and how the effort of model identification can be justified. MPC itself is a general term that includes a wide range of control methods that make explicit use of model to minimise a criterion function. The various MPC control techniques have following common ideas:

- Explicit use of a model for process prediction at future time instants.
- Computation of optimal control sequence.
- Receding horizon control principle where only the first sample of a computed control sequence is implemented.

The differences among various MPC algorithm are the model used to represent the process and the noise and cost function to be minimised. The earlier formulation of MPC favours finite impulse response (FIR) model and step response model. The corresponding algorithms that use FIR model include dynamic matrix control (DMC) (Cutler and Ramaker, 1980) and the quadratic DCM formulation of (Garcia and Morshedi, 1986). The FIR based MPC offers transparent description of process time delay, response time and gain. However, the FIR based MPC algorithm is not applicable to unstable plants and often requires large model orders depending on the process and sampling interval. Predictive control design using state space model (Maciejowski, 2002; Rawlings and Mayne, 2009) is also widely used as it can easily describe multivariable processes. Nevertheless, the state space representation of system with dead time would require an increased dimension of state vector especially in a digital system with a high sampling rate, hence high computing burden. The transfer function model on the other hand provides a parsimonious description of process dynamics and is not limited to stable plants. The transfer function

based MPC consists of predictive control algorithm by (Peterka, 1984) and the generalized predictive control (GPC) of (Clarke et al., 1987).

The MPC, in comparison with other control methods, presents a series of advantages (Camacho and Bordons, 2004) :

- The idea of MPC is intuitive and thus attractive to user with limited knowledge of control.
- MPC can be applied to a variety of processes, ranging from those relatively simple dynamics to the more complex systems with long dead time, non-minimum phase and unstable processes.
- It can be easily extended to multivariable case.
- It intrinsically accounts for dead times.
- Feed forward control can be incorporated to compensate for measurable disturbances.
- Treatment of constraints is easily included in the design process.
- Known future references are useful in computation of control sequence.
- The open methodology allows for future extensions.

The methodology of MPC family of controller can be characterised by the following strategy:

1. The future output sequence for an interval defined as prediction horizon, N is estimated at sample instant t based on process model. The predicted output $\hat{y}(t+k \mid t), \quad t \in \{1,..,N\}$ is computed using the known input output data up to sample instant t and the unknown future control signals $u(t+k \mid t), \quad k = \{0,...,N\}$ are calculated.
2. The future control sequences are to be computed so to minimise the error between the predicted outputs sequence and reference trajectory. The criterion is usually in the form of quadratic function of the prediction output error and incremental control signal.
3. The first element of the control sequence is sent to the process following the receding horizon concept. At the next sampling instant, $\hat{y}(t+1)$ becomes available and the procedure repeats itself from step 1 to calculate for the best control signal.

The overall MPC idea is shown in Figure 37.

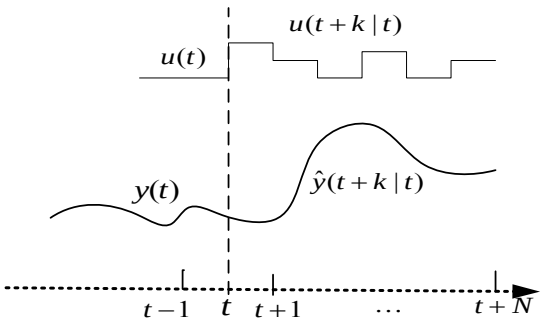


Figure 37: MPC concept.

Two different approaches of model prediction control are studied. First, generalised predictive control (GPC) scheme that belongs to the family of model predictive control is implemented to control the heading of the AUV helicopter. As the helicopter attitude dynamics model used in the controller design is in linear form, the designed controller has to guarantee stability and performance in the presence of large aerodynamics variations, neglected dynamics and non-linearity.

On the other hand, the Nonlinear MPC (NMPC) could be an alternative predictive control that explicitly addresses non-linearity. However, theoretical and practical issues remain unsolved, especially for applications that require fast control algorithm. A model predictive neural control is presented in (Wan and Bogdanov, 2001) where network feedback controller is combined with a state-dependent Riccati equation controller. However, the corresponding algorithm computation is apparently too heavy to be employed as real-time controller for UAV helicopter.

In GPC control scheme, a certain degree of tolerance is assumed towards model uncertainties that exist in the nominal model. The controller is then designed in such a way

that satisfactory performances are guaranteed in spite of the presence of model parameter variations or more generally the variations of dynamics characteristic of the helicopter.

However, the range of uncertainty domain that the robust controller could achieve depends very much on the closed-loop dynamics of the system. Thus in order to assure a satisfactory performances for a larger domain of uncertainty or full range of possible parameters variation, adaptive control is to be considered. The underlying robust controller is combined with a robust parameter estimator with respect to unmodelled dynamics and external disturbances to provide an applicable adaptive controller. In addition, the tuning in the adaptive control for the true nominal model parameter also improves the achieved performance of the robust controller design.

Another kind of model predictive scheme presented is based on model prediction and disturbance identification. The method uses continuous-time model in the design of the underlying predictive controller, and develops a control scheme that is robust towards the unmodelled dynamics and external disturbances.

In this chapter, details of the GPC control scheme are presented in Section 6.2 and the Dead-time Compensator (DTC)-based GPC with increased robustness is described in Section 6.3. Adaptive GPC control is presented in Section 6.4, while the model based control with disturbance identification is discussed in Section 6.5. Section 6.6 gives the summary of this chapter.

6.2 Generalized Predictive Control (GPC)

In the 1980s, adaptive control methods such as minimum variance adaptive control had been widely used in process control. However, these adaptive techniques rely on

accurate models, which might not be available for complex process. Thus their applicability has been limited to a certain degree. GPC developed along the adaptive control inherits the properties of adaptive control, specifically for its applicability in stochastic systems, while preserving the advantages of predictive control (Bao-Cang, 2010). The use of GPC as the helicopter attitude control is motivated by the its insensitivity with respect to plant model mismatch and offset-free performance capability. Furthermore, the applicability of GPC to industrial processes is well validated through successful applications reported in (Clarke, 1988) and an in-depth stability analysis in optimal control context can also be found in (Bitmead et al., 1990).

6.2.1 Prediction Model

The model used in GPC algorithm is a Controlled Auto-Regressive Integrated Moving Average (CARIMA) model given by (Clarke et al., 1987) in the form:

$$A(z^{-1})y(t) = z^{-d}B(z^{-1})u(t-1) + \frac{C(z^{-1})e(t)}{D(z^{-1})\Delta} \tag{6.1}$$

where $\Delta = 1 - z^{-1}$, d is the dead time, $e(t)$ is a white noise zero mean value. The integrator is added to ensure offset-free steady state control. The polynomials $C(z^{-1})$ and $D(z^{-1})$ represent the stochastic characteristics of the disturbance of the process. They are commonly set to $\mathrm{C}(z^{-1}) = D(z^{-1}) = 1$ as it is difficult for CARIMA model to fully describe the deterministic disturbance and in some cases time-varying characteristic of the disturbances. However, the polynomial $C(z^{-1})$ can be useful in the tuning procedure to enhance the robustness of the controller as shown in (Rossiter, 2013).

Consider the common setting of polynomials $C(z^{-1})$ and $D(z^{-1})$ being equal to one, an equivalent model of system in (6.1) is

$$\tilde{A}(z^{-1})y(t) = z^{-d}B(z^{-1})\Delta u(t-1) + e(t) \tag{6.2}$$

where

$$\begin{aligned}\tilde{A}(z^{-1}) &= A(z^{-1})(1-z^{-1}) = 1-\tilde{a}_1 z^{-1}\ldots-\tilde{a}_{na+1}z^{-na-1} \\ &= 1-(1-a_1)z^{-1}-(a_1-a_2)z^{-2}\ldots-a_{na}z^{-na-1}\end{aligned}$$

In order to deduce the prediction of future output sequence $y(k+j\,|\,k)$ with data available up to k sample instant, the Diophantine equation is considered:

$$1 = E_j(z^{-1})A(z^{-1})\Delta + z^{-j}F_j(z^{-1}) \tag{6.3}$$

where $E_j(z^{-1})$,.. are polynomials uniquely determined by $A(z^{-1})$ and length j,

$$\begin{aligned}E_j(z^{-1}) &= e_{j,0} + e_{j,1}z^{-1} + \ldots + e_{j,j-1}z^{-(j-1)} \\ F_j(z^{-1}) &= f_{j,0} + f_{j,1}z^{-1} + \ldots + f_{j,na}z^{-na}\end{aligned}$$

Multiplying (6.1) by $E_j(z^{-1})\Delta z^j$ and utilizing (6.3), the prediction at $k+j$ is:

$$\hat{y}(k+j\,|\,k) = E_j(z^{-1})z^{-d}B(z^{-1})\Delta u(k+j-1) + F_j(z^{-1})y(k) + E_j(z^{-1})e(k+j) \tag{6.4}$$

Since at time instant k, the future noises $e(k+i), i \in \{1,\ldots, j\}$ are unknown, thus the output prediction definition can be represented by (Clarke et al., 1987):

$$\hat{y}(k+j\,|\,k) = E_j(z^{-1})z^{-d}B(z^{-1})\Delta u(k+j-1) + F_j(z^{-1})y(k) \tag{6.5}$$

Let the convolution $E_jB = \mathrm{G}_j(z^{-1})$ and has a degree order of

$$\begin{aligned}dG_j &= dE_j + dB \\ &= j-1+dB\end{aligned} \tag{6.6}$$

Therefore, $G_j(z^{-1})$ can be rewritten as

$$G_j = g_{j,0} + \ldots + g_{j,dG_j} z^{-(j-1+dB)} \tag{6.7}$$

In order to split the $\Delta u(k+j-1)$ into past input samples, $\Delta u(k-i), i \in \{1,\ldots,n_b\}$ and undetermined future control sequences, $\Delta u(k+i), i \in \{0,\ldots,j-1\}$, $G_j(z^{-1})\Delta u(k+j-1)$ in (6.5) are to be reorganised and recursively computed for $j = 1 \ldots N$.

$$\begin{aligned} & G_j(z^{-1})\Delta u(k+j-1) \\ &= \begin{bmatrix} g_{1,0} & 0 & \ldots & 0 \\ g_{2,1} & g_{2,0} & & 0 \\ & & & \vdots \\ g_{N,\mathrm{N}} & g_{N,N-1} & & g_{N,0} \end{bmatrix} \begin{bmatrix} \Delta u(k) \\ \Delta u(k+1) \\ \vdots \\ \Delta u(k+N) \end{bmatrix} + \begin{bmatrix} g_{1,1} & \cdots & g_{1,dG_j} \\ g_{2,2} & & g_{2,dG_{j+1}} \\ \vdots & & \vdots \\ g_{N,N+1} & \cdots & g_{N,dG_{j+N}} \end{bmatrix} \begin{bmatrix} \Delta u(k-1) \\ \Delta u(k-2) \\ \vdots \\ \Delta u(k-N+1-dB) \end{bmatrix} \end{aligned} \tag{6.8}$$

Finally, another Diophantine equation is introduced:

$$E_j(z^{-1})B(z^{-1}) = \tilde{G}_j(z^{-1}) + z^{-(j-1)}H_j(z^{-1}) \tag{6.9}$$

where
$$\begin{aligned} \tilde{G}_j(z^{-1}) &= g_{j,0} + g_{j,1}z^{-1} + \ldots + g_{j,j-1}z^{-(j-1)} \\ H_j(z^{-1}) &= h_{j,1}z^{-1} + \ldots + h_{j,2}z^{-2} + \ldots + h_{j,n_b}z^{-n_b} \end{aligned}$$

Then, applying (6.9) in (6.5) yields

$$\begin{gathered} \hat{y}(k+j \mid k) = \tilde{G}_j(z^{-1})\Delta u(k+j-1-d) + H_j(z^{-1})\Delta u(k-d) + F_j(z^{-1})y(k) \\ j \in \{1,\ldots,N\} \end{gathered} \tag{6.10}$$

where the future prediction output is obtained based on future inputs, known inputs and measured outputs following the sequence of terms in (6.10).

6.2.2 Solution to Diophantine Equation

In order to solve (6.3), (Clarke et al., 1987) proposed an iterative algorithm to compute the polynomials $E_j(z^{-1}), F(z^{-1})$. From (6.3), following two equations are obtained,

$$\begin{aligned} 1 &= E_j(z^{-1})A(z^{-1})\Delta + z^{-j}F_j(z^{-1}) \\ 1 &= E_{j+1}(z^{-1})A(z^{-1})\Delta + z^{-j+1}F_{j+1}(z^{-1}) \end{aligned} \tag{6.11}$$

Subtracting first equation from second equation in (6.11) yields

$$A(z^{-1})\Delta[E_{j+1} - E_j] + z^{-j}[z^{-1}F_{j+1}(z^{-1}) - F_j(z^{-1})] \tag{6.12}$$

By denoting $E_{j+1}(z^{-1}) - E_j(z^{-1}) = \tilde{E}_j(z^{-1}) + e_{j+1,j}z^{-j}$, then, (6.12) becomes

$$\tilde{A}(z^{-1})\tilde{E}(z^{-1}) + z^{-j}[z^{-1}F_{j+1}(z^{-1}) - F_j(z^{-1}) + \tilde{A}(z^{-1})e_{j+1,j}] = 0 \tag{6.13}$$

In order to satisfy the equation of (6.13), $\tilde{A}(z^{-1})\tilde{E}(z^{-1})$ should be equal to zero. Since the first term in $\tilde{A}(z^{-1})$ is 1, it is necessary that $\tilde{E}(z^{-1}) = 0$. In addition, the second term in (6.13) should be equated to:

$$F_{j+1}(z^{-1}) = z[F_j(z^{-1}) - \tilde{A}(z^{-1})e_{j+1,j}] \tag{6.14}$$

Comparing the terms of the same order on both sides of (6.14) yields:

$$\begin{aligned} e_{j+1,j} &= f_{j,0} \\ f_{j+1,i} &= f_{j,i+1} - \tilde{a}_{i+1}e_{j+1,j} = f_{j,i+1} - \tilde{a}_{i+1}f_{j,0}, \quad i \in \{0,\ldots,n_a - 1\} \\ f_{j+1,n_a} &= -\tilde{a}_{n_a+1}e_{j+1,j} = -\tilde{a}_{n_a+1}f_{j,0} \end{aligned}$$

The formulation for deducing coefficients of $F_{j+1}(z^{-1})$ can be written in the form of a vector as follows:

$$f_{j+1} = \tilde{A}(z^{-1})f_j \tag{6.15}$$

and can be expanded into

$$\begin{bmatrix} f_{j+1,0} \\ f_{j+1,1} \\ \vdots \\ f_{j+1,\mathrm{n}_a} \end{bmatrix} = \begin{bmatrix} 1-a_1 & 1 & 0 & \cdots & 0 \\ a_1-a_2 & 0 & 1 & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & 0 \\ a_{n_a}- & 0 & \cdots & 0 & 1 \\ a_{n_a} & 0 & 0 & \cdots & 0 \end{bmatrix} \begin{bmatrix} f_{j,0} \\ f_{j,1} \\ \vdots \\ f_{j,\mathrm{n}_a} \end{bmatrix} \tag{6.16}$$

The iterative formula for coefficients of $E_{j+1}(z^{-1})$ is as follows:

$$E_{j+1}(z^{-1}) = E_j(z^{-1}) + e_{j+1,\mathrm{j}} z^{-j} = E_j(z^{-1}) + f_{j,0} z^{-j} \tag{6.17}$$

With the initialisation of $f_0 = [1, 0 \ldots 0]^T$, $E_0 = 0$, the coefficients of the polynomials $E_j(z^{-1}), F_j(z^{-1})$ $j \in \{1, \ldots, \mathrm{N}\}$ can be obtained recursively.

6.2.3 Formulation of Objective Function and Control Law Computation

The objective function of GPC is defined as:

$$J = \sum_{j=N_1}^{N_2} \delta(j)[\hat{y}(t+j \mid t) - w(t+j)]^2 + \sum_{j=1}^{N_u} \lambda(j)[\Delta u(t+j-1)]^2 \tag{6.18}$$

where $\hat{y}(t+j \mid t)$ is an optimum j-step ahead prediction output in data up to time t, $w(t+j)$ is the known future reference trajectory, $\Delta u(t+j-1)$ is the future control effort. $\delta(j), \lambda(j)$ are the weighting sequence for the quadratic function, and determine the extent of output error and control effort to be penalized. N_1, N_2 are the minimum and maximum costing horizons, N_u is the control horizon. N_1, N_2 are usually selected to be $N_1 = d+1$ and $N_2 = d+N$, where d is the time delay of system response, since the output samples earlier than N_1 are not affected by the control action to be computed from time sample t.

The objective of the predictive control is to minimise the error between the predicted output and the desired reference trajectory with the lowest possible control effort by finding the minimal J. Using (6.10) in prediction of output from $t+d+1$ to $t+d+N$ results in

$$\begin{bmatrix} \hat{y}(t+d+1|t) \\ \hat{y}(t+d+2|t) \\ \vdots \\ \hat{y}(t+d+N|t) \end{bmatrix} = \mathbf{G}\begin{bmatrix} \Delta u(t) \\ \Delta u(t+1) \\ \vdots \\ \Delta u(t+N) \end{bmatrix} + \mathbf{H}\begin{bmatrix} \Delta u(t-1) \\ \Delta u(t-2) \\ \vdots \\ \Delta u(t-nb) \end{bmatrix} + \mathbf{F}\begin{bmatrix} \hat{y}(t+d|t) \\ \hat{y}(t+d-1|t) \\ \vdots \\ \hat{y}(t+d-na|t) \end{bmatrix} \tag{6.19}$$

where

$$\mathbf{G} = \begin{bmatrix} g_{1,0} & 0 & \cdots & 0 \\ g_{2,1} & g_{2,0} & \ddots & \vdots \\ \vdots & \vdots & \ddots & 0 \\ g_{N_u,N_u-1} & g_{N_u,N_u-2} & \cdots & g_{N_u,0} \\ \vdots & \vdots & \ddots & \vdots \\ g_{N,N-1} & g_{N,N-2} & \cdots & g_{N,N-N_u-1} \end{bmatrix}, \quad \mathbf{H} = \begin{bmatrix} h_{1,1} & \cdots & h_{1,nb} \\ \vdots & \ddots & \vdots \\ h_{N,1} & \cdots & h_{N,nb} \end{bmatrix}, \quad \mathbf{F} = \begin{bmatrix} f_{1,0} & \cdots & f_{1,na} \\ \vdots & \ddots & \vdots \\ f_{N,0} & \cdots & f_{N,na} \end{bmatrix}$$

As can be seen from (6.19), additional output prediction from $\hat{y}(t+d-na)$ to $\hat{y}(t+d)$ is needed unlike predictive control of system without time delay which only requires the known outputs. The prediction follows the free response concept where future control sequences are kept constant and equal to last known control signal, $u_f(t+j) = u(t-1), \quad j \in \{0,\ldots,d\}$ (Camacho and Bordons, 2004).

The vector form of (6.19) can be written as

$$\hat{\mathbf{y}} = \mathbf{G}\mathbf{u} + \mathbf{H}\mathbf{u}' + \mathbf{F}\mathbf{y}' \tag{6.20}$$

where $\mathbf{y}', \mathbf{u}'$ are the output predictor value and the known past input respectively, $\mathbf{u}$ is the control action to be computed. Substituting (6.20) in (6.18) leads to

$$J = (\mathbf{Gu} + \mathbf{Hu}' + \mathbf{Fy}' - \mathbf{w})^T \mathbf{Q}_\delta (\mathbf{Gu} + \mathbf{Hu}' + \mathbf{Fy}' - \mathbf{w}) + \mathbf{u}^T \mathbf{Q}_\lambda \mathbf{u} \tag{6.21}$$

where $\mathbf{Q}_\delta$ and $\mathbf{Q}_\lambda$ are diagonal $N \times N$ weighting matrices with elements $\delta(j)$ and $\lambda(j)$, respectively. After some manipulations J becomes

$$\begin{aligned} J = {} & \mathbf{u}^T (\mathbf{Q}_\lambda + \mathbf{G}^T \mathbf{Q}_\delta \mathbf{G})\mathbf{u} + 2(\mathbf{Hu}' + \mathbf{Fy}' - \mathbf{w})\mathbf{Q}_\delta \mathbf{Gu} \\ & + (\mathbf{Hu}' + \mathbf{Fy}' - \mathbf{w})^T \mathbf{Q}_\delta (\mathbf{Hu}' + \mathbf{Fy}' - \mathbf{w}) \end{aligned} \tag{6.22}$$

and minimisation of J which is $\dfrac{\partial J}{\partial \mathbf{u}} = 2\mathbf{u}^T (\mathbf{G}^T \mathbf{Q}_\delta \mathbf{G} + \mathbf{Q}_\lambda) + 2\mathbf{G}^T (\mathbf{Hu}' + \mathbf{Fy}' - \mathbf{w}) = 0$ results in optimal increment control action of

$$\mathbf{u}^T = (\mathbf{G}^T \mathbf{Q}_\delta \mathbf{G} + \mathbf{Q}_\lambda)^{-1} \mathbf{G}(-\mathbf{Hu}' - \mathbf{Fy}' + \mathbf{w}) \tag{6.23}$$

or equivalently

$$\Delta u(t) = m\mathbf{Du}' + m\mathbf{Ny}' + m\mathbf{Pw} \tag{6.24}$$

where m is the first row of matrices $\mathbf{M} = (\mathbf{Q}_\lambda + \mathbf{G}^T \mathbf{Q}_\delta \mathbf{G})^{-1}$ of dimension $N \times N$, since only the first element of incremental control signals of $\mathbf{u}$ is needed following the receding horizon concept. $\mathbf{N} = -\mathbf{G}^T \mathbf{Q}_\delta \mathbf{F}$ of dimension $N \times (na + 1)$, $\mathbf{D} = -\mathbf{G}^T \mathbf{Q}_\delta \mathbf{H}$ of dimension $N \times nb$, $\mathbf{P} = \mathbf{G}^T \mathbf{Q}_\delta$ of dimension $N \times N$. The GPC control scheme of a process with a time delay is shown in Figure 38.

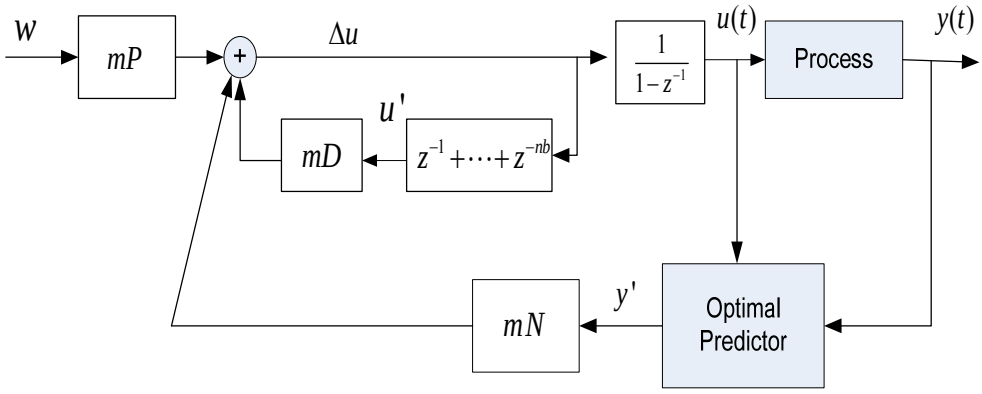


Figure 38: GPC control scheme.

6.2.4 Robustness Analysis of GPC

For robustness analysis purpose, the GPC controller is being derived into 2DOF structure. First, the control increment in (6.23) is rewritten as

$$[1-lu_1z^{-1}-lu_2z^{-2}-\ldots-lu_{nb}z^{-nb}]\Delta u(t)=ly_1\hat{y}(t+d\,|\,t)+\ldots+ly_{na+1}\hat{y}(t+d-na\,|\,t)+\sum_{i=1}^{N}f_i w(t+d+i)$$

(6.25)

where $[ly_1\ ly_2\ldots ly_{na+1}]=mN$, $[lu_1\ldots lu_{nb}]=mD$ and $\left[f_1\ldots f_N\right]=mP$. The predictions are substituted with

$$\begin{aligned}\hat{y}(t+d\,|\,t)&=F_d(z^{-1})y(t)+B(z^{-1})z^{-1}E_d(z^{-1})\Delta u(t)\\&\vdots\\\hat{y}(t+d-na\,|\,t)&=F_{d-na}(z^{-1})y(t)+B(z^{-1})z^{-na-1}E_{d-na}(z^{-1})\Delta u(t)\end{aligned}\qquad(6.26)$$

and after manipulations (Normey-Rico, 2007), the compact form is presented as

$$\begin{aligned}&\left[D_c(z^{-1})+\frac{G(z^{-1})}{\Delta}N_c(z^{-1})-\frac{P(z^{-1})}{\Delta}N_{fr}(z^{-1})\right]\Delta u\\&=N_{fr}(z^{-1})y(t)+N_f(z^{-1})w(t)\end{aligned}\qquad(6.27)$$

where $G(z^{-1})$ is the model of the plant without dead time, $G(z^{-1}) = \dfrac{z^{-d}B(z^{-1})}{A(z^{-1})}$. $P(z^{-1})$ is the complete process model, $P(z^{-1}) = \dfrac{B(z^{-1})z^{-1}z^{-d}}{A(z^{-1})}$. The rest of polynomial are:

$N_f(z) = \sum_{i=1}^{N} f_i z^{d+i}$, $D_c(z^{-1}) = [1 - lu_1 z^{-1} - lu_2 z^{-2} - \ldots - lu_{nb} z^{-nb}]$, $N_c(z^{-1}) = -\sum_{i=1}^{na+1} ly_i z^{-i+1}$,

$N_{fr}(z^{-1}) = -\sum_{i=1}^{na+1} ly_i F_{d-i+1}(z^{-1})$. The control equivalent equation can then be formulated as:

$$C_{eq}(z^{-1}) = \frac{u(t)}{y(t)} = \frac{\dfrac{N_{fr}(z^{-1})}{D_c(z^{-1})\Delta}}{\left[1 + \dfrac{G(z^{-1})}{D_c(z^{-1})\Delta} N_c(z^{-1}) - \dfrac{P(z^{-1})}{D_c(z^{-1})\Delta} N_{fr}(z^{-1})\right]} \tag{6.28}$$

Equation (6.28) can be interpreted as

$$C_{eq}(z) = \frac{C(z)F_r(z)}{1 + C(z)G(z)\left[1 - z^{-d}F_r(z)\right]} \tag{6.29}$$

with

$$\begin{aligned} C(z) &= \frac{N_c(z^{-1})}{D_c(z^{-1})\Delta} \\ F_r(z) &= \frac{N_{fr}(z^{-1})}{N_c(z^{-1})\Delta} \end{aligned} \tag{6.30}$$

Subsequently, the transfer function of input actions to reference trajectory can also be derived as

$$\frac{u(t)}{w(t)} = \frac{C(z)F(z)}{1 + C(z)G(z)\left[1 - z^{-d}F_r(z)\right]} \tag{6.31}$$

with

$$F(z) = \frac{N_f(z^{-1})}{N_c(z^{-1})} \tag{6.32}$$

By substituting (6.29) and (6.31) in (6.27), the resulting equation is

$$u(t) = \frac{C(z)}{1 + C(z)G(z)\left[1 - z^{-d}F_r(z)\right]}\left[F(z)w(t) + F_r(z)y(t)\right] \tag{6.33}$$

and the equivalent structure that describes the Equation (6.33) is shown in Figure 39. The GPC is in fact has a similar structure to dead time compensator (DTC) in the form of Smith Predictor (SP) except that $F_r(z)$ term is not equal to one as in the standard SP structure. It can be seen from the structure that if $q(t) = 0$ and process $P(z) = G_n(z)z^{-d_n}$, the characteristic equation is

$$1 + C(z)G_n(z) \tag{6.34}$$

Therefore, the dead time of the process is eliminated from the closed loop characteristic equation, and it does not impose further phase margin, which deteriorates the stability of the system. Consequently, larger tuning space is available for achieving tracking performance as desired.

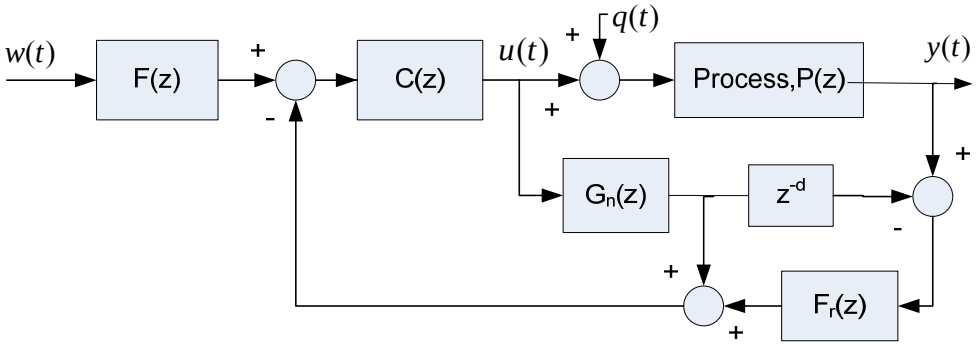


Figure 39: Equivalent structure of GPC.

Robustness analysis of the controller towards model/plant mismatch or simply model uncertainty is based on the block diagram of Figure 39. Model uncertainties can be grouped into two main classes, structured and unstructured uncertainties. Parametric uncertainty is quantified by a mean parameter value and a relative uncertainty in the parameter (Skogestad and Postlethwaite, 2005):

$$\alpha_p = \bar{\alpha}(1 + r_p \Delta) \tag{6.35}$$

where $\bar{\alpha}$ is the parameter mean value, r_p is the relative parameter uncertainty $r_p = (\alpha_{\max} - \alpha_{\min}) / (\alpha_{\max} + \alpha_{\min})$ assuming the uncertain parameter is bounded within a range of $[\alpha_{\max}, \alpha_{\min}]$, Δ is any real scalar satisfying $|\Delta| \leq 1$. The uncertainties represented this way will not be able to account for the residual modelling error that arises from unmodelled dynamics, especially in high frequency region (Doyle and Stein, 1981). This is because models tend to describe low frequency behaviour well but become inaccurate for high frequency inputs (Morari and Zafiriou, 1989). Therefore, parametric uncertainties are often described in the form of unstructured uncertainties by replacing the real perturbation with complex perturbation $|\Delta(j\omega)| \leq 1$. If several real perturbations exist, the generated uncertainties region in Nyquist plane has complicated shape and mathematical descriptions (Skogestad and Postlethwaite, 2005). Nevertheless, the complicated shape that is generated by varying the parameters (static gain, time constant and dead time) within certain bound is often quite "disc like" as illustrated in Figure 40. As a result, lumping these perturbations into a disc-shaped region as single complex perturbation can simplify the analysis and controller synthesis.

The disc shaped regions that represent the uncertainty regions can be represented in the form of additive uncertainty:

$$l_a(\omega) = \max_{G_p \in \Pi} \left| G_p(j\omega) - \tilde{G}(j\omega) \right| \tag{6.36}$$

Alternatively, it can also be represented as multiplicative uncertainty of the form:

$$l_m(\omega) = \max_{G_p \in \Pi} \left| \frac{G_p(j\omega) - \tilde{G}(j\omega)}{\tilde{G}(j\omega)} \right| \tag{6.37}$$

where Π is a set of possible perturbed plant model $G_p(s)$ in a vicinity of the nominal process $\tilde{G}(s)$. At each frequency ω, $l_a(\omega), l_m(\omega)$ represent the smallest radius of the disc-shaped region centred at $\tilde{G}(s)$, which include all possible plant as shown in Figure 40.

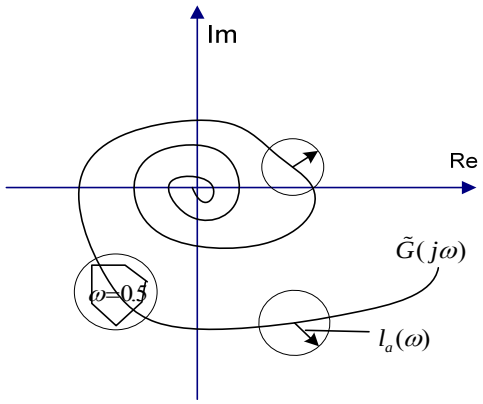


Figure 40: Parametric uncertainties as unstructured uncertainty.

Among the uncertainties considered are errors in estimation of gains, dead time, dominant time constant and unmodelled dynamics. The errors in estimation of dead time is the main factor that affects the frequency domain uncertainties from medium to high

frequencies range. On the other hand, the error in the gain and time constant have dominant effect in low frequencies (Normey-Rico, 2007).

Robust stability is a phenomenon that the system remain stable in the face of all perturbations in the uncertainty set. According to robust stability theorem of (Morari and Zafiriou, 1989), an uncertain feedback system as shown in Figure 41 with $C(z)$ being the controller and $\tilde{G}$ being the nominal process, is to possess robust stability if and only if, the complementary sensitivity function $T(s)$ for the nominal process fulfils the relation:

$$\left|T(s)l_m\right| < 1 \tag{6.38}$$

This can be derived from Nyquist stability condition, where according to the stability condition the stability of the closed loop system in Figure 41 is guaranteed if the loop transfer function does not encircle point -1. With uncertainty the loop transfer function becomes:

$$L_p = G_p C = \tilde{G}C(1 + l_m) \tag{6.39}$$

It follows then that the distance from -1 to the nominal loop transfer function $\tilde{G}C$ is equivalent to $\left|-1 - L_p\right| = \left|1 + L_p\right|$. The encirclement of point -1 is avoided if the uncertainty disc of radius $\left|\tilde{G}Cl_m\right|$ does not cover -1, ie:

$$\begin{aligned} \left|\tilde{G}Cl_m\right| &< \left|1 + \tilde{G}C\right|, \forall \omega \\ \left|\frac{\tilde{G}C}{1 + \tilde{G}C}\right| &< \frac{1}{\left|l_m\right|} \end{aligned} \tag{6.40}$$

Note that Equation (6.40) coincides with the inverse of the system closed-loop transfer function for $F(z)=1$, which is the complementary sensitivity function $T(s)$.

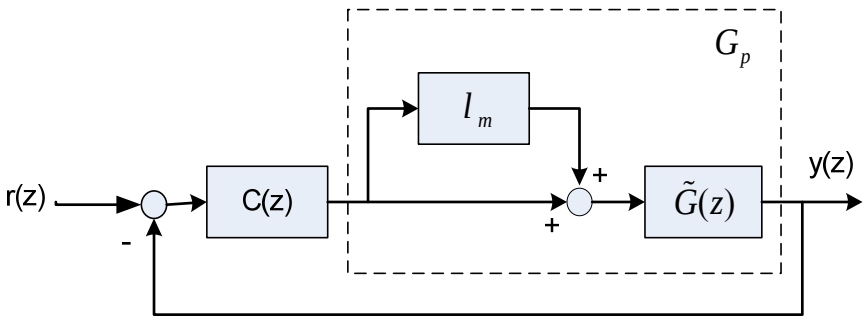


Figure 41: Process and controller with unity feedback.

Before the same principle can be applied to GPC, the GPC controller structure has to be transformed into a structure that similar to Figure 41, so that the complementary sensitivity function $T(s)$ can be obtained easily. The resulting GPC equivalent structure is illustrated in Figure 42 and is in fact a compact form of block diagram from Figure 39. As there is an extra block element of $F(z)$ in GPC structure, the robust stability analysis of the GPC can be tackled using the above theorem as long as block F is proven stable. It is shown in (Camacho and Bordons, 1995) that the denominator polynomial of rational transfer function of block $F(z^{-1})$ always has one term more than that of the numerator, thus quotient is always less than one and the root is bound within unit circle. Therefore, it is safe to state that $F(z^{-1})$ is stable and does not affect the results about the stability of the closed loop.

Following derivation of Equation(6.29), Ceq in Figure 42 is denoted as:

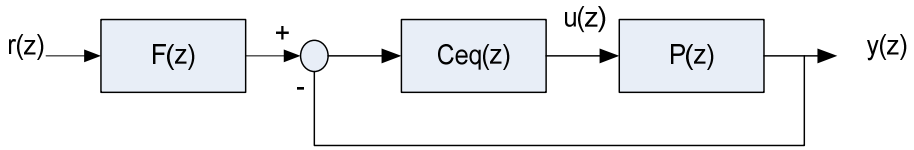


Figure 42: GPC block diagram.

$$Ceq = \frac{C(z)F_r(z)}{1 + C(z)\left(G_n(z) - P_n(z)F_r(z)\right)} \tag{6.41}$$

and the complementary sensitivity function $T(z)$ can be obtained from Figure 42 with reference to Figure 41 as:

$$T(z) = \frac{C(z)P(z)F_r(z)}{1 + C(z)G_n(z)} \tag{6.42}$$

The robust condition can then be formed:

$$\left|\delta P(z)\right| < l_m = \frac{\left|1 + C(z)G_n(z)\right|}{\left|F_r(z)C(z)G_n(z)\right|} \tag{6.43}$$

where $\delta P(z)$ is the uncertainty in the model and it has to be below the multiplicative uncertainty limit of l_m which in turn depends system time delay. Since l_m is inverse proportional to filter $F_r(z)$,

$$F_r(z) = \frac{ly_1 F_{d_n}(z^{-1}) + ly_2 F_{d_n-1}(z^{-1}) + \ldots + ly_{na+1} F_{d_n-na}(z^{-1})}{ly_1 + ly_2(z^{-1}) + \ldots + ly_{na+1} z^{-na}}, \tag{6.44}$$

the higher the value of system time delay, F_{d_n} also increases accordingly and lowers the uncertainty limit. This means that GPC has different robustness characteristics for different nominal dead times, d_n. Thus, a process with higher nominal dead time is easily driven into instability compared with a process with smaller nominal dead time. High sampling

rate also affects the robustness stability, as the discrete time delay sample, d_n becomes high value even when the system dead time is low.

6.3 DTC-Based GPC

In order to improve the robustness of GPC, T-polynomial is normally introduced as a fixed observer or prefilter. The prefilter is shown to play an essential role in the robust realisation of predictive controllers (Clarke and Mohtadi, 1989). The prefiltering of T-polynomial is shown to be a particular case of Q-parameterisation (Yoon and Clarke, 1995). The Q-parameterisation can be obtained through minimisation of H_∞ norm, which can be computationally demanding especially in the adaptive case. Although the prefilter is implied to be an alternative to the optimal Q, intuitive rule for T-tuning does not exist as has been explained in (Yoon and Clarke, 1995), where stronger filtering does not necessarily yield better robustness properties in presence of high-frequency unmodelled dynamics. The stronger filter here means that the filter bandwidth is narrower and rolls off more rapidly than the other. This is due to the complex relationship between the T-polynomial and the robustness condition dP_{GPC}. Therefore, the approach of using the predictor of dead-time compensator (DTC) is considered explicitly in the computation of the free response of the GPC strategy so that the robustness in the basic GPC (T polynomial equal to 1) can be improved.

It is noted that in Figure 38, the computation of the controller is done independent of the predictor structure, thus different predictor structure can be used. The realisation of SP-based GPC is shown in Figure 43, where the optimal predictor is replaced with SP

structure for predictions up to $t+d$, while primary controller is obtained with GPC procedure.

In order to further improve the robustness stability of SP-GPC, a filter can be included in predictor block (Normey-Rico, 2007; Normey-Rico et al., 1999). The resulting filtered SP-GPC has a similar structure to GPC as in Figure 39. In SP structure, the filter F_r is tuning block that improves the robustness stability of closed loop control in frequency region where the modelling uncertainty is significant. It is different from F_r in GPC where its computation is determined by system dead time and controller parameters. To further elaborate the implementation details, assuming zero value for the undetermined future input, the output predictions from $t+1$ up to $t+d$ are computed using equation:

$$\begin{aligned} \hat{y}(t+1\,|\,t) &= G_n(z^{-1})u(t+1) + F_r(z)[y(t) - P_n(z^{-1})u(t+1)] \\ &\vdots \\ \hat{y}(t+d\,|\,t) &= G_n(z^{-1})u(t+d) + F_r(z)[y(t+d-1) - P_n(z^{-1})u(t+d)] \end{aligned} \tag{6.45}$$

By substituting (6.45) in (6.19) for the output $\hat{y}(t+d)\cdots\hat{y}(t+d-na)$, and following the normal procedure of GPC, the control actions from $\Delta u(t)...\Delta u(\mathrm{t}+Nu)$ can then be computed using (6.23), with the coefficients of $m\mathbf{D}, m\mathbf{N}, m\mathbf{P}$ in (6.24) being derived from optimal prediction Equation (6.10).

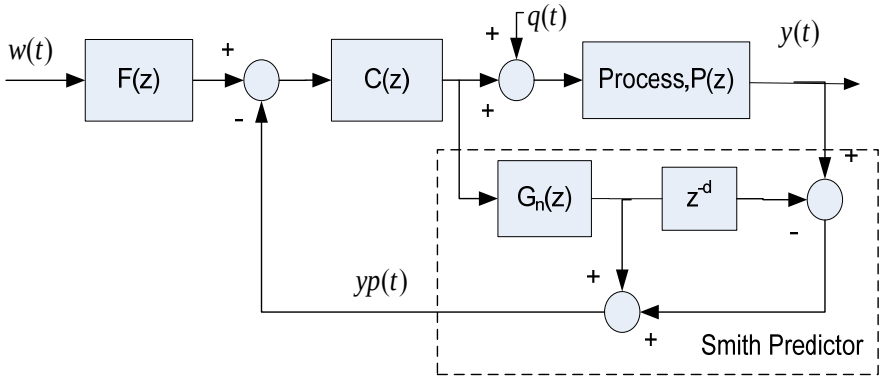


Figure 43: Smith predictor based GPC.

The properties of filtered SP-GPC can be summarised as follows. Firstly, with the same choice of $C(z)$ and $F(z)$ as in GPC, the filtered SP-GPC has the same set-point nominal performance as GPC since it is independent of $F_r(z)$ and is shown as

$$H_r(z) = \frac{F(z)C(z)G_n(z)z^{-d_n}}{1+C(z)G_n(z)} \tag{6.46}$$

and the multiplicative uncertainty limit $l_m(\omega)$ can be computed as:

$$l_m(\omega) = \frac{|1+C(z)G_n(z)|}{|F_r(z)C(z)G_n(z)|} \tag{6.47}$$

On the other hand, the closed-loop disturbance transfer function for filtered SP-GPC can also easily be obtained from Figure 39 as follows:

$$H_q(z) = G_n(z)z^{-d_n}\left[1-\frac{F_r(z)C(z)G_n(z)z^{-d_n}}{1+C(z)G_n(z)}\right] \tag{6.48}$$

where the transfer function is F_r dependent. This implies that F_r has to be tuned to achieve a compromise between disturbance rejection response performance and the enhancement of robust stability of the closed loop control. Whereas, the set point tracking

performance is decoupled from the disturbance rejection performance as it is controlled solely by $F(z)$, which is the characteristic of 2-DOF controller structure.

6.3.1 Robust DTC-Based GPC Tuning

As the SP structure is adopted in the GPC, some of the inherent drawbacks from SP are also present in SP-GPC, and filter F_r plays an important role in circumventing the problems while providing a higher bandwidth for the controller robustness. First, in the basic SP structure without filter, the disturbance rejection transient cannot be arbitrarily defined because the poles of $P(s)$ cannot be eliminated in the transfer function of input disturbance to output. From the equivalent structure of SP in Figure 44, the transfer function can be written as:

$$\frac{Y(s)}{Q(s)} = \frac{P_n(s)}{1 + P_n(s)C_{eq}(s)} \tag{6.49}$$

with

$$C_{eq}(s) = \frac{C(s)}{1 + C(s)(G_n(s) - P_n(s))} \tag{6.50}$$

Considering $G_n(s) = \frac{N_p(s)}{D_p(s)}, P_n(s) = G_n(s)e^{-L_n s}$, it is obvious that D_p is a factor in numerator of $C_{eq}(s)$. Thus, $C_{eq}(s)$ has zeros in the same position as the plant poles $P_n(s)$, and after pole-zero cancellation in (6.49), the closed loop poles of $\frac{Y(s)}{Q(s)}$ have the plant poles. The same result is applied even in the case where $C(s)$ cancels the plant poles. This property of SP implies that the pole of the plant dominates the transient response of

disturbance rejection. However, it only has important effect when the plant has slow time constant, then the disturbance rejection transient response appears to be very sluggish. Even worse is when the process has an unstable pole that cannot be eliminated from the disturbance rejection transfer function. In this case, the disturbance rejection may have gone into instability.

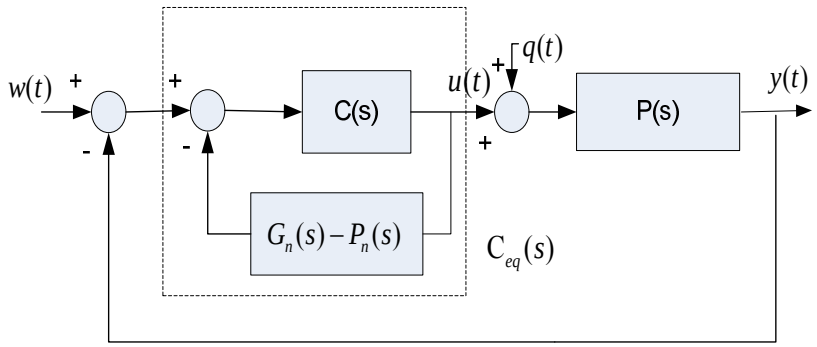


Figure 44: Equivalent SP structure.

The second drawback is that in its basic form, SP is not able to be used with integrative processes as it will give an offset response when input disturbance is present. As can be seen from Figure 44, the primary controller has a feedback block of $H(s) = G_n(s) - P_n(s)$. Assume a nominal plant of $P_n(s) = \frac{P_e}{s}$, where $P_e(s) = G_e(s)e^{-L_n s}$ and is stable. To reject a constant step disturbance $\frac{1}{s}$, according to the internal model principle the equivalent controller $C_{eq}(s)$ must have an integral action (a pole at $s = 0$). This can be achieved provided $H(s)$ in (6.50) is equal to zero and an integrator is included in $C(s)$. However, it will be shown later that this is not possible in an integrative system. Due to the

existence of integrator in the system model, the static gain of $\lim_{s->0} H(s)$ becomes indeterminate and has to be computed with L'hopital rules:

$$\lim_{s\to 0} H(s) = \lim_{s\to 0} G_e(s)\left[\frac{1-e^{-L_n s}}{s}\right] = L_n G_e(0) \neq 0 \tag{6.51}$$

This implies that even if $C(s)$ has an integral action, such as $C(s) = \frac{K_c}{s}$, the $C_{eq}(s)$ does not have the pole at $s = 0$,

$$\lim_{s\to 0} C_{eq}(s) = \lim_{s\to 0}\left[\frac{\frac{K_c}{s}}{1+H(0)\frac{K_c}{s}}\right] = \lim_{s\to 0}\left[\frac{K_c}{s+H(0)K_c}\right] = \frac{1}{H(0)} \tag{6.52}$$

Subsequently, the steady state gain of $\frac{Y(s)}{Q(s)}$ after substitution of (6.52) in (6.49) becomes:

$$\lim_{s\to 0}\left[\frac{P_n(s)}{1+P_n(s)C_{eq}(s)}\right] = \lim_{s\to 0}\left[\frac{P_e(s)}{s+P_e(s)C_{eq}(s)}\right] = \lim_{s\to 0}\frac{1}{C_{eq}(s)} \tag{6.53}$$

It is obvious then that the steady state value of $y(t)$ has a constant offset from set-point in face of step disturbance imposed on the system.

Therefore, in order to solve the above problems facing basic SP, one of the options is to use filtered SP and the corresponding structure of Figure 39. The corresponding equivalent controller is:

$$C_{eq}(s) = \frac{C(s)}{1+C(s)(G_n(s)-P_n(s)F_r(s))} \tag{6.54}$$

while the disturbance rejection transfer function is as follows:

$$\frac{Y(s)}{Q(s)} = \frac{P_n(s)}{1+P_n(s)C_{eq}(s)F_r(s)} \tag{6.55}$$

Pole zero cancellation between the zeros of $C_{eq}(s)$ and the poles of the nominal plant $P_n(s)$ might result in slow transient and instability on disturbance rejection response if the plant have slow time constant and unstable poles respectively. One way to eliminate the effects is to impose a zero at $s=0$ in the feedback block of the equivalent controller $C_{eq}(s)$ so that $H(s)|_{s=0} = G_n(s)(1-e^{-L_n s}F_r(s))|_{s=0} = 0$. Subsequently, $F_r(s)$ can be tuned to obtain desired disturbance rejection transient.

In addition, in order to have an offset free steady state disturbance rejection for an integrative process, the steady state gain of $\frac{Y(s)}{Q(s)}$ must be set to zero, such that:

$$\lim_{s\to 0}\left[\frac{P_n(s)}{1+P_n(s)C_{eq}(s)F_r(s)}\right] = \lim_{s\to 0}\left[\frac{P_e(s)}{s+P_e(s)C_{eq}(s)F_r(s)}\right] = \lim_{s\to 0}\frac{1}{C_{eq}(s)F_r(s)} = 0 \quad (6.56)$$

This can be done by imposing a condition on the feedback block $H(s)$ such that:

$$H(s) = \lim_{s\to 0}\frac{(1-e^{-L_n s}F_r(s))}{s} = 0 \quad (6.57)$$

Then, L'Hopital rule is used to evaluate the limits to obtain:

$$\lim_{s\to 0}\frac{d}{ds}(1-e^{-L_n s}F_r(s)) = 0 \quad (6.58)$$

This configuration can be generalised to reject disturbance of type steps, ramps, etc of the form $\frac{1}{s^m}$ in the steady state. Thus, the numerator of $1-e^{-L_n s}F_r(s)$ must contain m roots in $s=0$, and $F_r(s)$ is tuned as an m-order filter according to the type of disturbance, where $m=1$ for steps, $m=2$ for ramps, and etc, so to fulfil the condition $\frac{d}{ds^m} = 0$.

For implementation purposes, the discretised version of $F_r(s)$ is used. $F_r(s)$ that is tuned for an integrative process with step disturbance rejection must meet all the following conditions:

$$\left\{ \begin{array}{l} (F_r(z) - z^d)|_{z=1} = 0 \\ \dfrac{d}{dz}(F_r(z) - z^d)|_{z=1} = 0 \end{array} \right\} \tag{6.59}$$

A possible $F_r(z)$ is a second order filter of the form (Normey-Rico, 2007):

$$F_r(z) = \frac{f_{b0}z + f_{b1}}{(z-\beta)^2} \tag{6.60}$$

After substitution of (6.60) in (6.59), f_{b1} and f_{b0} are obtained as:

$$\begin{array}{l} f_{b1} = (1-\beta)^2 - f_{b0} \\ f_{b0} = (1-\beta)^2 d_n + 2(1-\beta) \end{array} \tag{6.61}$$

where β is the free parameter.

6.4 Adaptive GPC

It is often advantageous to change the controller to compensate for changes in plant parameters caused by age, wear, temperature, fatigue, etc. Ignoring the model mismatch that always exists in physical systems may create problems for the model-based control system. The advent of self-tuning and adaptive control in 1970s are meant to overcome these problems and provide tight control of processes despite stochastic disturbances, dead-time and parameters variations (Clarke and Mohtadi, 1989).

In the context of adaptive control, there are two main categories of adaptive control designs, namely the direct and indirect adaptive control. The direct adaptive control

scheme includes those methods in which parameters of the controller are directly updated from an adaptation error that reflects the discrepancy between the desired and the real output response. The direct adaptive control scheme that belongs to this categories includes Model Reference Adaptive Control (MRAC) (Landau, 1979), where the desired output is generated by a reference model. While the performance of the feedback control system can in many case be specified in terms of a reference model, the conditions for the existence of a feasible controller that allows the closed-loop to match the reference model is restrictive. The basic limitation is the assumption that all the plant zeros have to be stable which is restrictive in discrete-time case (Landau et al., 2011). For instance, high frequency sampling of continuous-time systems with difference of degree between the numerator and denominator is larger or equal to two leads to unstable zeros (Åström et al., 1984).

On the other hand, in indirect adaptive control, the adaptation of the controller is done in two stages where the plant parameters are estimated online initially, and followed by the computation of controller parameters based on the estimated plant model parameters. The design of adaptive GPC falls into this category. Open loop recursive parameter identification, specifically the RPEM in Section 4.4.2, is used to adjust the model parameters such as $\hat{a}$ and $\hat{b}$ through minimizing the error between plant output and predicted output. Using certainty equivalent principle, the online estimated model parameters are treated as if they are equal to the true parameters. The estimated model parameters are used to compute the controller parameters $m\mathbf{P}, m\mathbf{D}$ and $m\mathbf{N}$ via adaptation mechanism. The prediction of output response $y' = \left[\hat{y}(t+d \mid t) \cdots \hat{y}(t+d-na \mid t)\right]$ is

obtained by the use of the SP as described in Section 6.3. The proposed control scheme is shown in Figure 45.

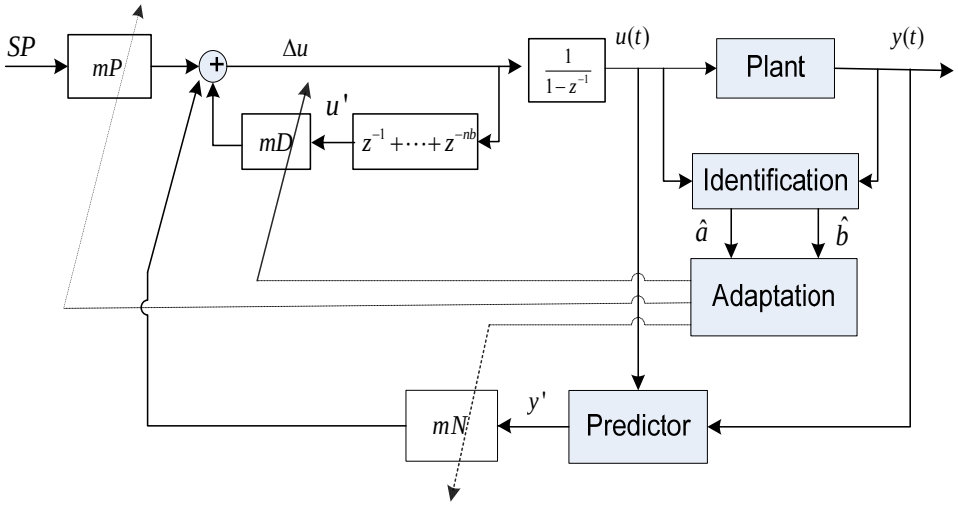


Figure 45: Adaptive GPC control scheme.

6.5 Model Based Control with Disturbance Identification

A control scheme based on model prediction and disturbance identification has been developed. The model prediction used in the control design is in the form of continuous-time model. Although the model and design are based on continuous model, the controller is implemented in a digital system, which offers advantages to systems with a fast sampling rate.

6.6 Summary

In this chapter, the standard GPC and the SP-based GPC are presented to fulfil the control objectives of achieving satisfactory robust stability margins while ensuring the desired tracking performances. The controller uses a prediction model that explicitly accounts for the dead-time in dynamic system and is equally feasible for reduced order system dynamics. The certainty-equivalence-based adaptive version of SP-based GPC is also presented in the perspective of performance improvement and extension of the range of operation. As compared with the nonlinear MPC control scheme, the proposed control algorithm only requires small computation time, thus is suitable for real-time application such as UAV helicopter. Lastly, a control scheme that is based on a continous model prediction was developed. This method utilisers the iterative integral method to identify disturbance and predicts forward with analytical expressions so is very computationally efficient.

7. UAV HELICOPTER CONTROL RESULTS

7.1 Introduction

For the ease of carrying out testing on heading control, the heading angle control results are obtained based on the constraining the pitch and roll axis freedom of movement of the helicopter to simulate hovering condition. This is justified as tail thrust response to a blade pitch change in small-scale UAV can be considered quasi-instantaneous and this response is much faster than the fuselage dynamics response, hence the yaw-channel dynamics can be physically decoupled from other channels in hover and near-hover conditions (Mettler, 2003; Hyunchul Shim et al., 2000).

In order to obtain constant main rotor torque, the experiments are carried out with the collective pitch and throttle maintained at a constant value. I.e. the normalised collective pitch δ_{col} is set to 0.5 and throttle is set to 50%. When testing the control algorithm, the external control command generated is passed through the yaw rate gyro, which has been described in Section 3.2, instead of being a direct input to the helicopter system. The yaw rate gyro used is Align GP790 heading-lock gyro with an exclusive digital servo DS420. As the update rate from the gyro is generally higher than the receiver, it has to be paired with a high speed digital servo to work properly. The microprocessor in the digital servo analyses the received signal and converts this into high frequency voltage pulses to the servo motor, which enables fast motor turning and maintains constant torque so that the tail rotor can stabilise the highly sensitive yaw channel on the helicopter. The block diagram of the control configuration is shown in Figure 46.

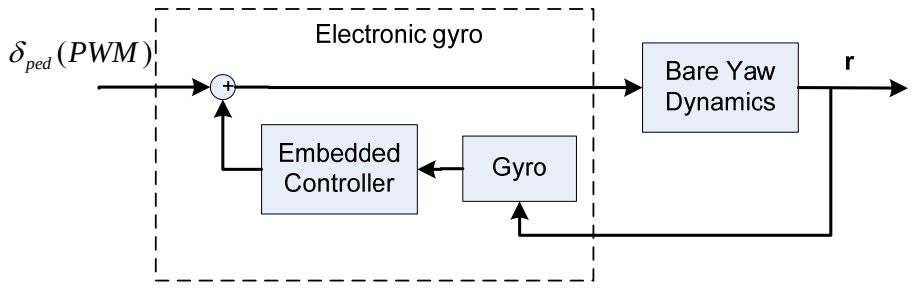


Figure 46: Configuration of yaw channel of the UAV helicopter.

Due to time constraint, the roll and pitch control are implemented with PD control. Since the lateral and longitudinal cyclic pitch control signals are translated into three PWM servo control signals, therefore, both the pitch and roll control signal have to be computed simultaneously in the controller.

7.2 Implementation Issues

7.2.1 Actuators Non-Linearity

A high precision control system requires that all the components, ie the sensor, the controller and the actuators work properly. However, in reality the performance of a control loop is always limited by the actuator non-linearity, such as hysteresis, backlash, dead-band, dead zone and stiction. Stiction is the most harmful among the nonlinearities, leading to unsatisfactory performance. According to (M.Ruel, 2000), stiction severity of 1% or more is likely to cause considerable oscillations in controlled response. The stiction severity is generally measured as the percentage of the span of the control signal.

Stiction phenomenon is observed as the resistance of the actuator to move under a small control changes. When the control change is larger than a specific limit that move the actuator output, the actuator output makes an abrupt jump over the expected target. Then

the control change has to increase gradually until it is large enough to move the actuator output in the opposite direction. This cycle repeats itself about the set target. Because of stiction, the actuator output will not move smoothly and this applied indirectly to the manipulated variable.

The harmful effect of stiction has been an area of study, where researchers are faced with challenges such as modelling, detection, quantification and compensation. Existing stiction modelling can be classified into two major groups, which are the physics-based (first principle) and data driven models. Data driven models are commonly used due to their relative simplicity. On contrary the physics-based models require numerous parameters and variables to be estimated and measured, which limits the application of these models. The stiction phenomenon to be modelled is shown in Figure 47. A simple model is used in this work, which assumes that staticband (deadband+stickband, S) and slip jump (J) are equal. This model is described by (Garcia, 2008) :

$$x(t) = \begin{cases} x(t-1) & \text{if } \left|u(t) - u_{x(t-1)}\right| \le d \\ u(t) & \text{otherwise} \end{cases} \tag{7.1}$$

where $x(t-1)$ and $x(t)$ are the past and present servo actuator position respectively. $u(t)$ is the actual controller output and $u_{x(t-1)}$ is the controller output when the actuator position first appears to freeze at the same position. d is the servo actuator stiction band. The one parameter model is considered here because only the measurement of the stiction band is needed, and is adequate for designing the compensating signal in closed loop. Since the measurement of servo position is usually unavailable, the stiction band d can be estimated

from the closed loop data as the peak to peak value of the controller output (Gerry, 2001). This represents a type of stiction detection method through time domain shape analysis.

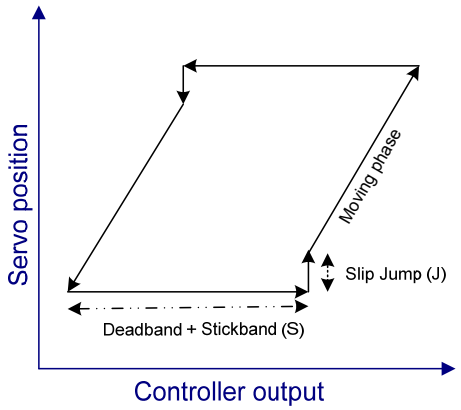


Figure 47: Relation between controller output and servo position.

The stiction phenomenon causes a limited resolution of the resulting control servo actuator motion. Certainly, this problem can be solved by replacing the servo actuator with servo actuator that has high precision or lower stiction band. However, there is occasion when this solution cannot be considered a primary choice, then stiction compensation has to be implemented. The stiction compensation method presented in (Hägglund, 2002) can be considered as the first algorithm to tackle the problem, which mainly targets static friction in control valves. The idea of the compensation is to add a predesigned signal to controller output in the direction of the rate of change of the control signal in order to minimise the fluctuations of the process output as is depicted in Figure 48. The compensating signal is in the form of a sequence of pulses with constant amplitude a, width τ and period between two pulses h. Each pulse interval is given by:

$$u_c(t) = \begin{cases} a\ sign(\Delta u) & t \le t_p + h + \tau \\ 0 & t > t_p + h + \tau \end{cases} \tag{7.2}$$

where t_p is the time of onset of the previous pulse. Another similar idea of compensation is introduced by (Xiang Ivan and Lakshminarayanan, 2009) where the difference is that the additive signal is not of pulse form, but a constant value.

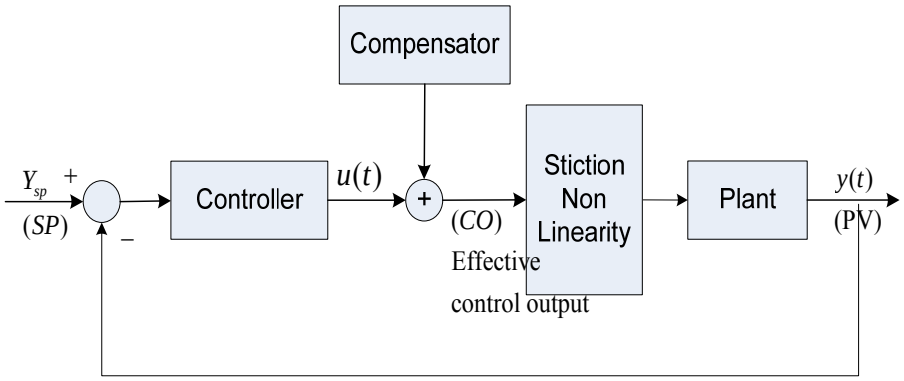


Figure 48: Control loop with compensator.

The stiction phenomenon can be observed from the shape of the output response as shown in Figure 49. The stiction band is approximately 5%. Using the constant reinforcement compensation method with $a = 2.5\%$ on the same controller setting, the result is shown Figure 50, where the oscillation in output response is eliminated. Although the output variability is reduced significantly, this reduction is achieved at cost of aggressive servo actuation movement. Therefore, knocker pulse method is used with knocker parameter $a = 2.5\%$, $\tau = 5$ and $h = 20$ to obtain a compromise between the output variability and the actuation movements.

The compensating signal introduces much larger controller output than controller output without compensating scheme. This creates an undesired effect when the actuator is in moving phase, especially during process state transition when the controller is catching up with the reference and at the same time has to compensate for the added stiction compensating signal. The resulting process output in this case appears choppy and high overshoot instead of smooth step transition. Therefore, a condition is added to only apply stiction compensation when the output error between the desired and measured value falls below some small threshold, which limiting the stiction compensation in regulating stage. The resulting input-output behaviour is shown in Figure 51.

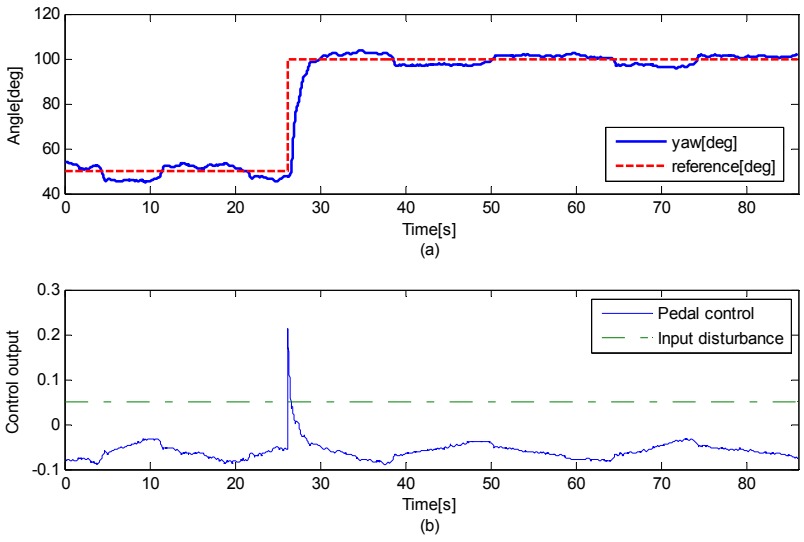


Figure 49: (a) Yaw angle response (b) Controller output.

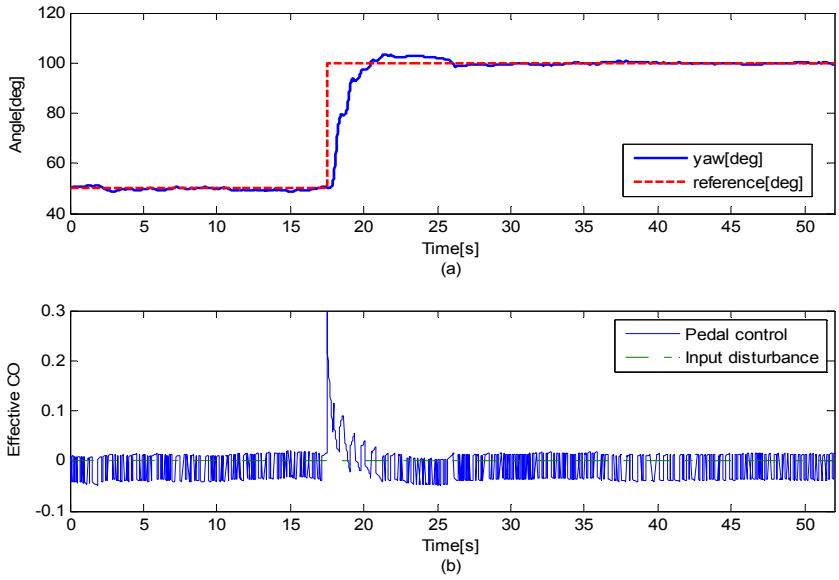


Figure 50: (a) Yaw angle response after constant reinforcement compensation (b) Effective controller output.

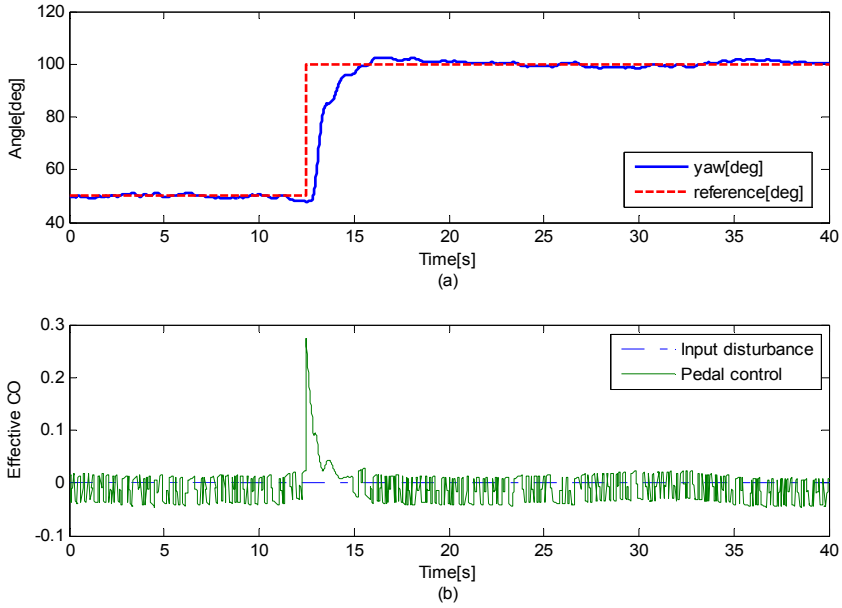


Figure 51: (a) Yaw angle response with conditional compensation (b) Effective controller output.

7.3 DTC-based GPC Control

In order for GPC control to be implemented as a real-time application, the controller is designed in discrete domain. For instance, the continuous-time yaw dynamics in Equation (4.24)can be rewritten in the form Laplace transform as:

$$P(s) = \frac{\psi(s)}{\delta_{ped}(s)} = e^{-sL}\,\frac{k_{10}}{s(s+k_{11})} \tag{7.3}$$

Equation (7.3) is basically a second order integral model with dead time and k_{10}, k_{11} is the parameters that were identified from system identification. The constant bias term $k_{d,3}$ in (4.24) is left out and is considered step input disturbance. Considering the dead time to be integer multiple of the sampling time, the corresponding discrete-time transfer function can be obtained using the discretisation function *c2d(.)* in MATLAB and to yield:

$$\mathrm{P}(z) = \frac{b_0 z^{-1} + b_1 z^{-2}}{1 - a_1 z^{-1} + a_2 z^{-2}}\, z^{-d} \tag{7.4}$$

In order to design GPC controller, CARIMA model is used to model the random disturbances in the system and the noise polynomial is chosen to be 1, the following equation is obtained:

$$(1 - a_1 z^{-1} + a_2 z^{-2})\psi(k) = (b_0 z^{-1} + b_1 z^{-2}) z^{-d} \delta_{ped}(k) + \frac{\varepsilon(k)}{\Delta} \tag{7.5}$$

where k is denoted as the discrete time sample. $\psi(k)$ and $\delta_{ped}(k)$ are the output and control sequences, $\varepsilon(k)$ is a zero mean white noise and $\Delta = 1 - z^{-1}$. The optimal estimation of $\hat{\psi}(k+j|k)$ can be computed following (6.10) and the GPC control law is computed by (6.24).

In order to compare the robustness of basic GPC and SP based GPC, a specific set of tuned controller parameters is used and the multiplicative uncertainty limit in expression (6.43) is plotted in frequency domain. The yaw dynamics model is considered with dead time of 0.4 *sec*, given controller sampling time $T_s = 0.02\text{sec}$, and $d = 20$. The basic GPC tuning parameters are configured as N=10, Nu=2, and λ=10. For the SP based GPC, the additional filter $F_r(z)$ is obtained from (6.60) with $\beta = 0.96$, thus:

$$F_r(z) = \frac{0.112z - 0.1104}{(z - 0.96)^2} \tag{7.6}$$

The simulation analysis for both cases as well as the modelling error that is solely due to dead time estimation error $\Delta d = 5$, i.e. $\delta P(\omega) = \left| e^{-j\Delta d\omega} - 1 \right|$, are shown in Figure 52. It can be seen that the robustness of SP-based GPC is significantly higher from medium to high frequency range compare to basic GPC. Besides, there also exists a major robustness deficiency in basic GPC when slight dead-time estimation error of 0.1s happens in the control system, whereas SP based GPC is immune to such parameter uncertainty.

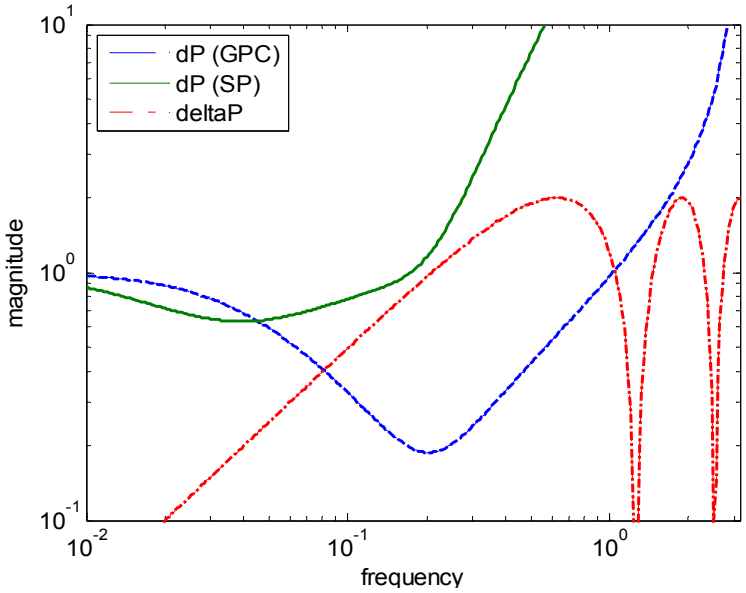


Figure 52: Uncertainty limit of GPC vs filtered SP-GPC.

The SP-based GPC is applied using the approximated model to control the heading of the helicopter. One of the main challenges in practical GPC is that there are a few parameters need to be tuned, mainly the costing horizon N, control horizon and control weighting. It is shown that the closed loop under GPC control is stable if the prediction horizon and control horizon tend towards infinity (Clarke and Mohtadi, 1989), as has been adopted in the infinite stage LQ controllers which are known to have good stability and robustness properties. However, in order to keep the computation of GPC within practical limits, the following design guidelines are recommended.

1) The first costing horizon at least exceeds the plant dead time.
2) N2 to be set $\geq 2n-1$, but would be better up to the rise time of the plant with n being the model order. If the computation power allowed, N2 can be set up to the settling time of the plant so that any possible non-minimum phase effect in the initial transients can be counteracted by ensuing positive-going behaviour.
3) $\mathrm{Nu} \leq n$.
4) $\lambda = \varepsilon$ where ε is small to help with numerical robustness.

Model predictive control is specifically easy to tackle the constraint control problem, and normally quadratic programming approach (Luenberger, 1984; Fletcher, 1981) is needed especially for a multi-input and multi-output system. As for a single-input and single-output system, with the receding horizon principle where only the first sample of a sequence of control trajectories is implemented, the optimal solution is obtained if the first sample is an active constraint. The rest of the constraints affect the optimal solution if they are activated, and the first sample of the constraint is not active. Therefore, from practical perspective, it is possible to only impose a constraint on the first sample, assuming the rest of constraints are not active or have little effect on the closed-loop performance (Wang, 2009).

In the following experiments, the closed-form solution for the constraint on the control variable is adopted. Let the constraints on the control variables be expressed as $u^{\min} \le u(k) \le u^{\max}$. Whenever $u(k) \le u^{\min}$ or $u(k) \ge u^{\max}$, instead of finding optimal control variable that satisfies the active constraint which becomes equality constraint, the control variable is directly configured to $u(k) = u^{\min}$ and $u(k) = u^{\max}$ respectively. For the tail/rudder servo control, the full value of rudder control signal is not used as the total tail servo operating angle is commanded by approximately 30% of the normalised rudder control signal. Therefore, for the heading angle control, the actuator saturation limit of ± 0.3 is imposed on absolute input $u(t)$.

Comparison of three different prediction horizons is made and the input-output results are shown in Figure 53. In this experiment, the following controller objective design parameters, i.e. $Nu = 2, \lambda = 10, \beta = 0.97$ are used with three different prediction horizon,

N . The tracking performance for all the experiments are almost similar in terms of rise time and overshoot, except that the settling time for $N = 15$ is slightly slower than $N = 10, N = 8$ by about 2s. On the other hand, the control input for $N = 8$ is rather aggressive and oscillatory compared with $N = 10, N = 15$, hence results in less smooth output step transition response. The deterioration of performance is due to the model mismatch, i.e. low order model and the fast tracking dynamics such as step reference. The results suggest that the prediction horizon, N provides the compromise between the tracking performance and the robustness stability of the closed-loop control system and $N = 10$ is considered the optimum tuning that achieves the balance between both criterions.

Another experiment is carried out to study the effect of control horizon on the yaw step response. The result is shown in Figure 54 with following control objectives design parameters: $N = 15, \lambda = 10, \beta = 0.97$, and the comparison is made for $Nu = 2$ and 5. The results show a faster response time for higher Nu, but at the expense of reduced robustness that is illustrated in the oscillatory step transition. In general, the control horizon is kept small in order not to increase the dimension of the matrix involved in the control law calculations and thus the computation burden.

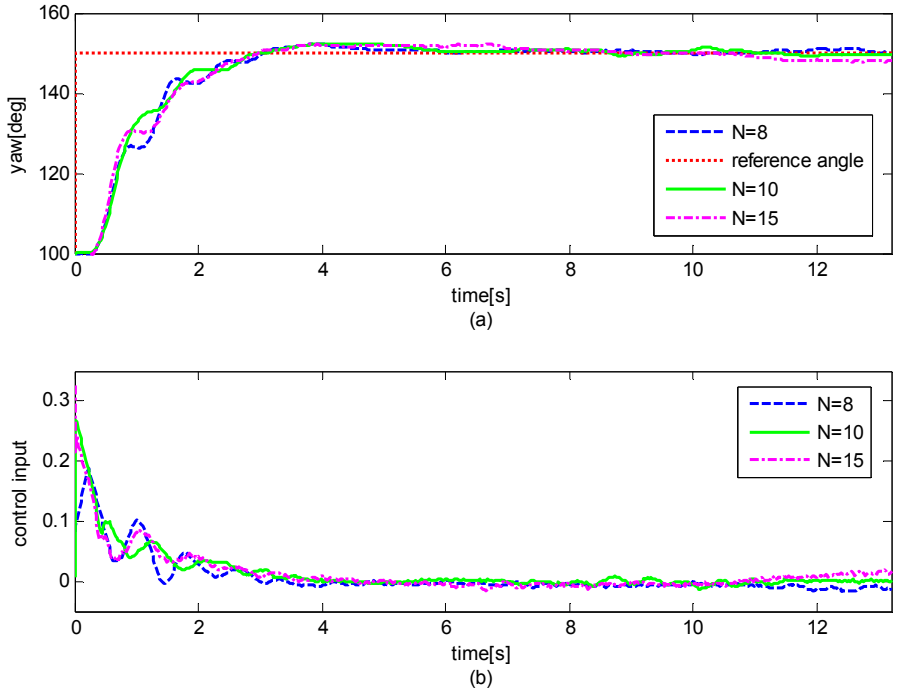


Figure 53: Effects of N on yaw step response: (a) Reference tracking output response (b) Controller output.

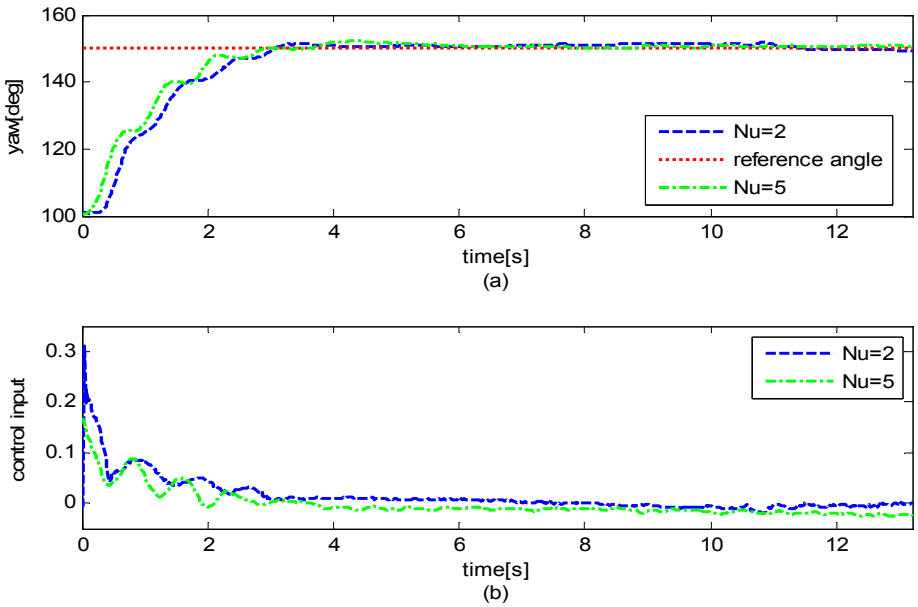


Figure 54: Effects of Nu on yaw step response: (a) Reference tracking output response (b) Controller output.

7.3.1 Effect of Smith Predictor (SP) Filter

In order to study the effect of filtering on the robustness and tracking performance, controller parameters are set as: predicton horizon, N=8, control horizon, Nu=2, and control weighting, $\lambda = 10$ for experiment with different filter parameter. The experiment result is shown in Figure 55, where the experiment with $\beta = 0.97$ exhibit better robustness towards the unmodelled dynamics and the varying model parameters than experiment with $\beta = 0.96$. The high sensitivity of the controller robustness towards the filter tuning parameter is expected due to the limited tolerance of uncertainty in the model. This is especially evident in the middle range frequency where normal operation of helicopter is likely to occur, as is shown in Figure 52.

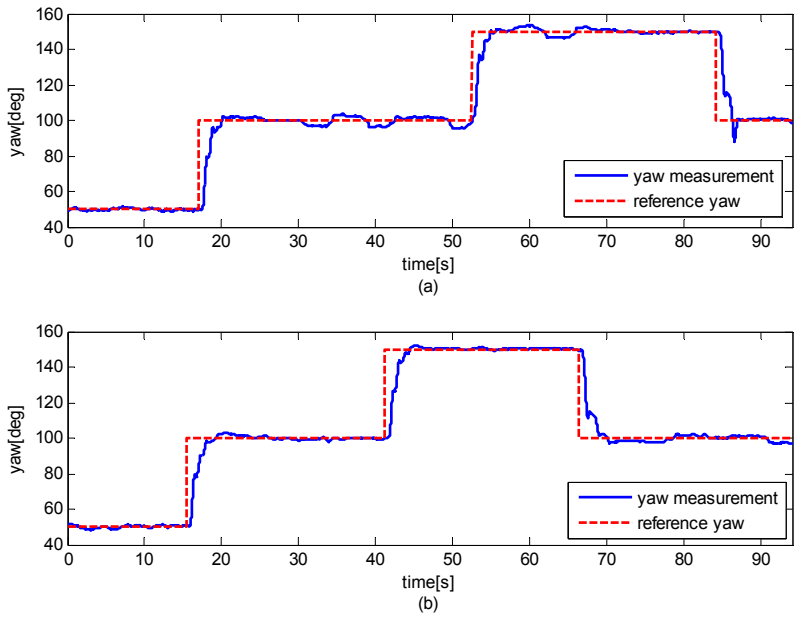


Figure 55: (a) filtering coefficent 0.96 (b)filtering coefficient 0.97

On the other hand, as has explained in Section 6.3, the tracking performance of both experiments are unaffected by the filter tuning parameter, β where both experiments have almost similar rise time and settling time of approximately 1s and 3s respectively as shown in Figure 56. However, the deterioration of robustness in the controller can be observed in the form of highly oscillatory step transition for the lower β value.

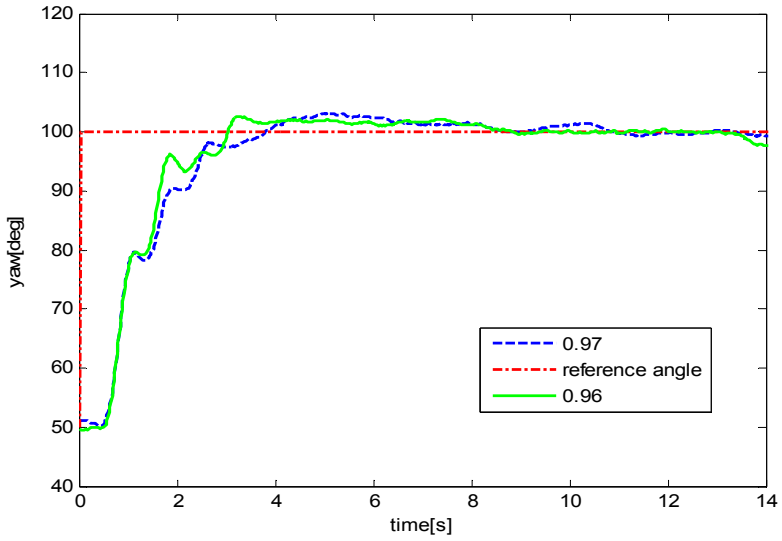


Figure 56: Zoom in on step response

As explain in Section 6.3, there is an inherent trade-off between the robustness and the disturbance rejection capabilities of the SP based GPC. Figure 57 shows the experimental result of disturbance rejection performance using different β value, i.e. $\beta = 0.96$ and 0.98. The controller output command and the step input disturbance are also shown in the second plot of Figure 57. It can be seen that higher β results in stronger filter magnitude $|Fr(z)|$. By following the transfer function describing the relationship between

output response and disturbance as shown in Equation (6.48), weaker disturbance rejection performance is implied. The overshoot of the output is about $3deg$ higher in $\beta = 0.98$ compare to that in $\beta = 0.96$, and the settling time is also a lot larger for larger β value which is approximation double the time taken while using lower β value.

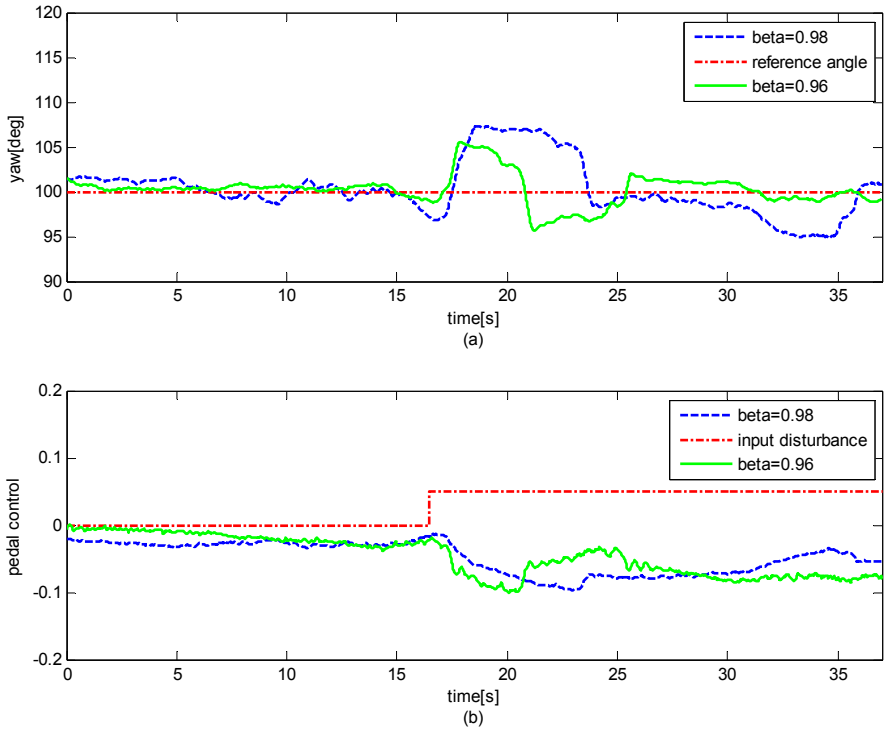


Figure 57: Step disturbance rejection performance for beta=0.96 and beta=0.98: (a) Yaw angle regulation (b) Controller output and input disturbance.

7.4 Comparison of GPC and GPC with Online Parameters Estimation

For comparison of GPC and adaptive GPC, the effect of parameters variation from performing high frequency step reference angle change can be observed from the robust GPC. The robust GPC are designed to cope with the parameters variation in the model in smaller range unable to perform perfectly with a fixed set of control objectives design parameters under the time-varying model parameter. On the other hand, the online parameters estimation constantly updates the parameters in the model to better approximate the real system. Thus, the control performance can be further improved.

The experimental evaluation for the adaptive GPC has been carried out according to the following experimental planning.

- The underlying robust GPC controller parameters are configured as: prediction horizon $N = 10$, control horizon $Nu = 2$, control weighting $\lambda = 10$ and filtering coefficient of $\beta = 0.98$.
- Each experiment is started in closed loop identification until the parameter adaptation is settles down, and then switches to adaptive control.
- As second order low pass Butterworth filter is used to filter input-output for the online parameter estimator. The cutoff frequency is set to be 5*Hz* which is chosen to be close to the closed loop bandwidth that is computed following the initial estimated model as discussed in Section 4.4.5.
- The scalar μ^2 used in the normalization factor is set to 0.95. This choice satisfies the condition such that the poles of discrete closed loop characteristic polynomial is bounded within μ, which is theoretically motivated (Giri et al., 1991).
- The adaptation algorithm is started with variable forgetting factor algorithm followed by constant trace algorithm. Various thresholds on the trace of the adaptation gain matrix *P(t)* and on the information measure are used to vary the adaptation gain matrix *P(t)* under different circumstances, and are set to 1.0 and 0.001 respectively from trial and error method. The adaptation

freezing is based on evaluation of the average of information measure over a sliding horizon of 30 samples.

Experiments are carried out on the yaw angle control to emphasize the applicability of the adaptive GPC controller and to compare it with the fix model GPC controller. The comparison mainly focuses on the performance of tracking capability and disturbance rejection. Figure 59 and Figure 58 show the input-output behaviour of the adaptive GPC and fix model GPC respectively. The output response has to follow a series of step reference on helicopter heading, and this step reference is considered a rich input excitation to the parameter estimator. The corresponding data information richness computed at the 100*deg* step reference at time instant 50 sec exceeds the threshold that trigger the scheduler to turn on the parameter adaptation, which is illustrated in Figure 60. Figure 61 reported the time constant parameter sequence $\{a_1(t)\}$ and $\{a_2(t)\}$. It shows that the time constant has a high variation at the initial stage, but remains almost unchanged afterwards. On the other hand, the parameter estimates sequence specifically the input gain constants $\{b_1(t)\}$ and $\{b_2(t)\}$ show obvious change in value during step reference change and under the presence of input disturbance, as can be seen from Figure 62. The changes in input constant during the step reference transition are due to the reaction force from the rotor system to the actuators, which is not accounted in the model. The filtering and parameter freezing appear to play a major role in avoiding parameter drifting in the estimator and thus guarantee the stability and performance of the adaptive controller.

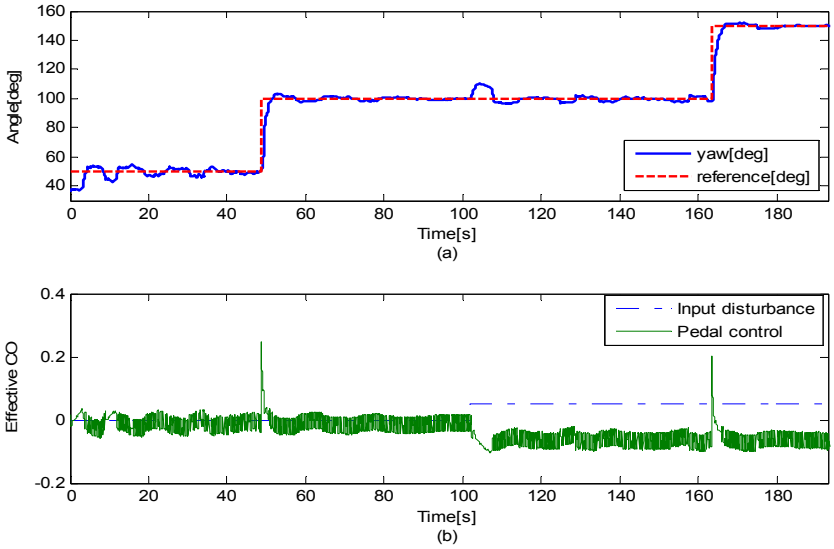


Figure 58: Fixed model GPC results: (a) Output response (b) Effective controller output.

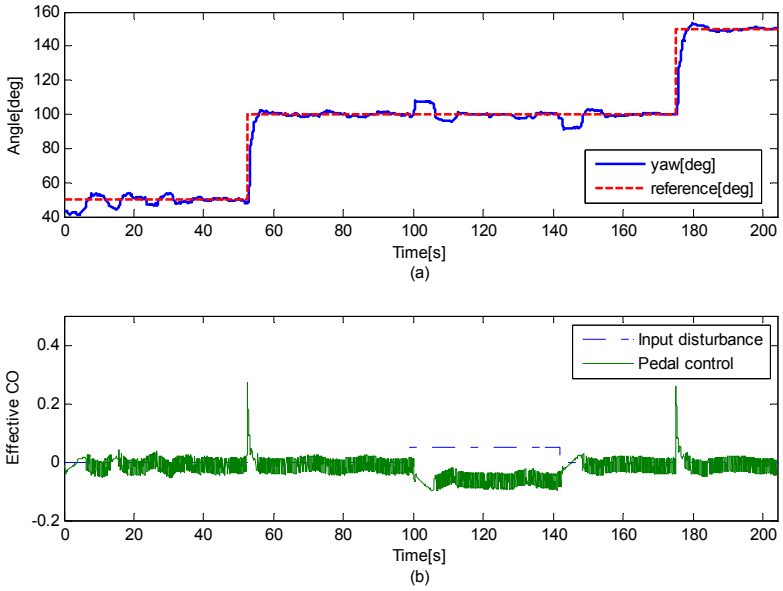


Figure 59: Adaptive GPC results: (a) Output response (b) Effective controller output.

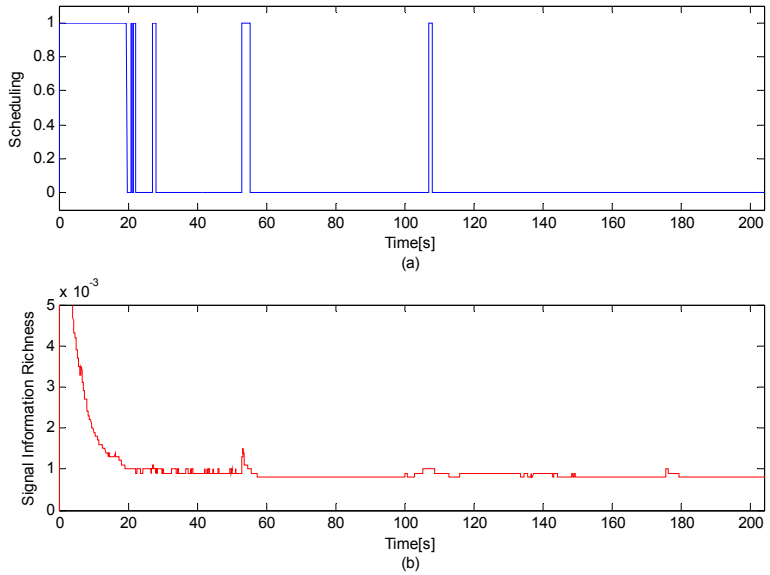


Figure 60: (a) Adaptation scheduling (b) Signal information richness.

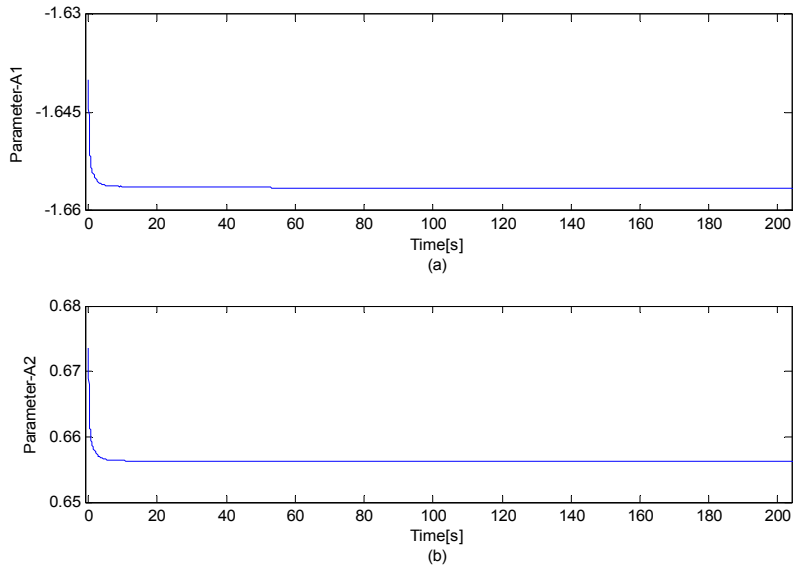


Figure 61: Estimated parameters: (a) $\hat{a}_1$ (b) $\hat{a}_2$.

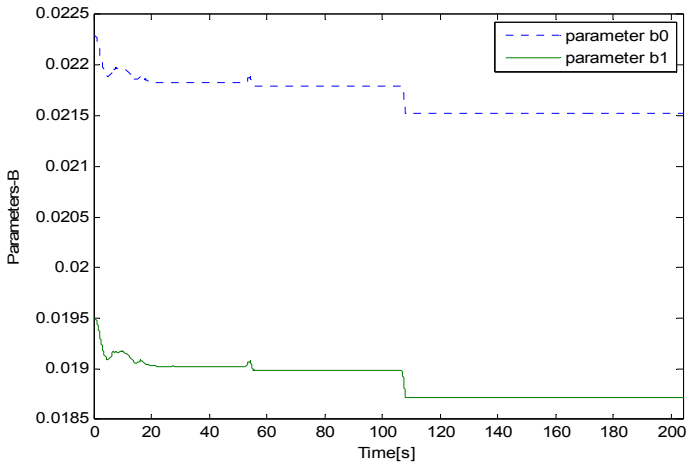


Figure 62: Estimated parameters b_0 and b_1.

7.4.1 Tracking Performance

The improvement of tracking performance can be seen from the zoom in view during the 100 deg step reference and is shown in Figure 63. Although the rise time is approximately the same for both the fix model GPC and adaptive GPC, the adaptive GPC has a slightly lower overshoot and a settling time of approximately 10 sec faster compared with the fix model GPC.

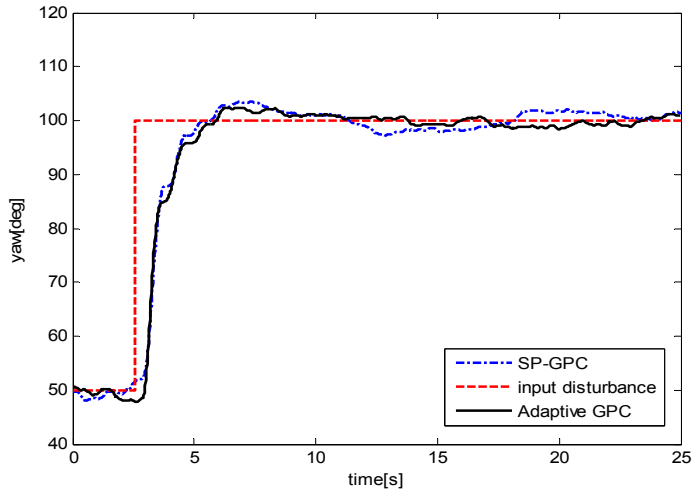


Figure 63: Zoom in view of step response at 100deg reference.

7.4.2 Adaptive GPC Disturbance Rejection Performance

A positive step disturbance of magnitude 0.05 is imposed on the system as input disturbance at time instant 100 sec. The disturbance rejection capability of both fix model GPC and adaptive GPC is compared in Figure 64, and the controller output is shown in Figure 65. The adaptive GPC exhibits a better disturbance rejection where the fast reaction of the control command to compensate the input disturbance resulting in lower overshoot compare with fix model GPC. Furthermore, the alertness of the parameter estimator towards input disturbance can be seen from Figure 62, where the input constant parameter shows a considerable change in value at the instant when input disturbance is purposely injected to the system.

It should be noted that the step input disturbance introduced is a realistic simulation of possible disturbance generation scenario due to sudden change of main rotor RPM and varying collective pitch command. Both actions create a change in the main rotor torque, which in turn affects the reactive torque on yaw channel of the helicopter.

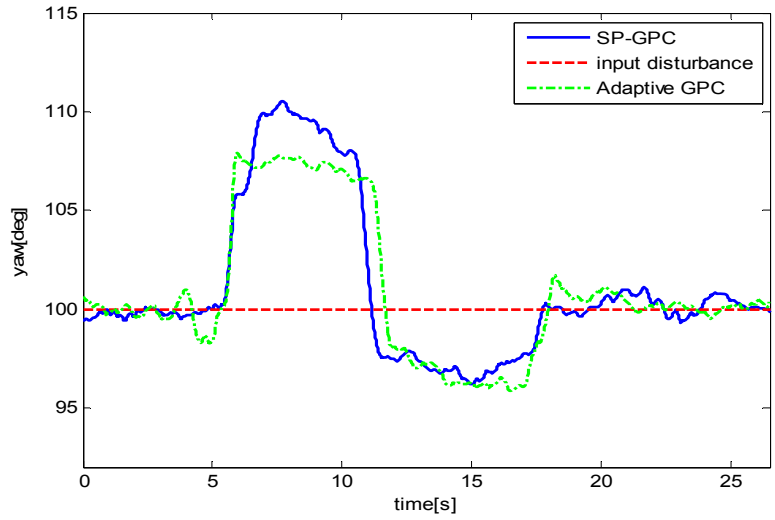


Figure 64: The disturbance response of SP-GPC vs adaptive GPC

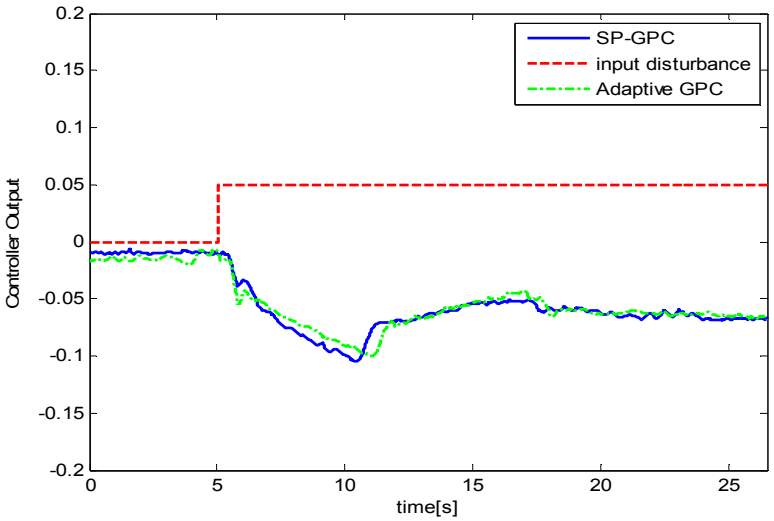


Figure 65: The control command of SP-GPC, adaptive GPC

7.4.3 Improvement of Performance for SP-Based GPC

In order to show the improvement of performance using adaptive GPC over DTC-based GPC towards the robustness with respect to the low order model and varying system parameters, a set of control objectives design parameters are chosen as follows:

$$N = 8, Nu = 2, \lambda = 10, \beta = 0.96 \tag{7.7}$$

When the control parameters are applied in DTC-based GPC, the deterioration of performance can be observed in the step response. The input-output results are shown in Figure 66. Such a result is expected since the filter tuning parameter is set to a relatively low value. On the other hand, when adaptive GPC is applied with the same control objectives design parameters for the underlying controller, smoother step transition is obtained.

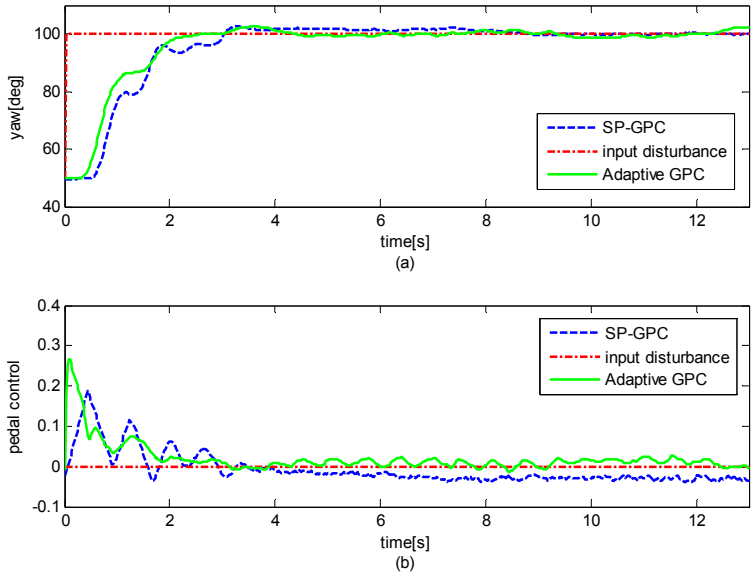


Figure 66: Robustness comparison between SP-GPC and Adaptive GPC: (a) Reference tracking output response (b) Controller output.

7.5 Model Based Control on Helicopter Heading

Model based control scheme as described in Section 6.5 is applied to the yaw axis of the helicopter for heading control. Transient reference angle is provided as the input to the control system and the output response is shown in Figure 67.

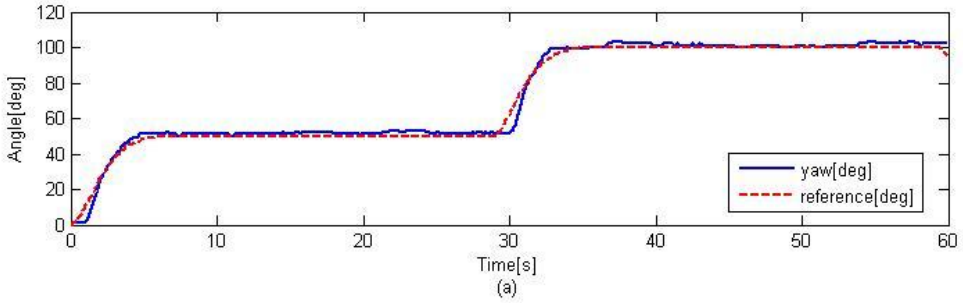


Figure 67: Model based control on yaw axis. Reference tracking output response

7.6 Helicopter Stabilisation in Hovering Condition

The stabilisation of row and pitch channel of helicopter in hovering condition is implemented using the improved PD control as discussed in Section 3.5 while the yaw channel is controlled by electronic gyro to maintain at zero angular speed. The control system architecture is shown in Figure 68. In the architecture, pitch, row, and yaw angle regulation are controlled separately as single input single output (SISO) system. This is justified with the finding of low coupling dynamics between the lateral and longitudinal channels in hovering and near hovering flight. The experiment starts with the helicopter

being spooled up at ground level using servomechanism as shown in Section 3.4, and then automatic control is switched on after certain RPM in the main rotor is achieved. After the steady state is achieved in the row-pitch channel, the supporting ground level platform is released to let the helicopter hovering on the test rig. Figure 69 shows the helicopter hovering on the test rig.

Figure 70 and Figure 71 show the row-pitch output response of the helicopter in hovering mode and the lateral-longitudinal command from the PD controller respectively. As PD controller cannot produce offset-free reference sequence tracking, the reference set point is set to compensate the bias output response. However, due to unavailability of exact prior knowledge about output response offset, the results obtained show minor offset in the output response from the expected zero degree of angular reference in row and pitch channel. Due to the offset, a perturbation moment is created by the gravitational attraction, which acts on the helicopter to cause the oscillation in the pitch and roll control response. Nevertheless, the stabilisation of the helicopter in row and pitch channel can be further improved if the offset in the output response can be eliminated.

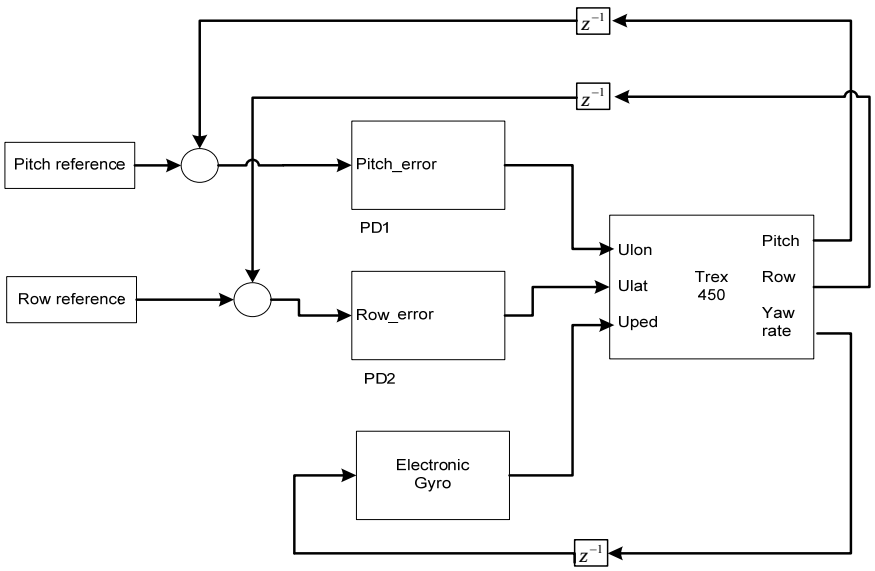


Figure 68: Control architecture for helicopter stabilization.

Figure 69: Helicopter performing hovering on test rig.

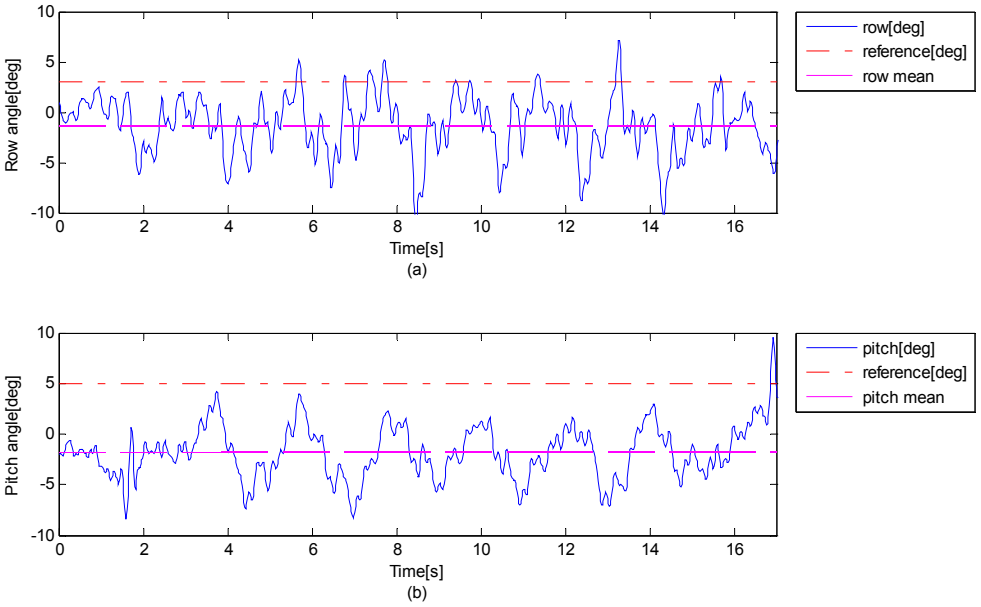


Figure 70: PD control result on (a) row angle and (b) pitch angle for helicopter hovering mode.

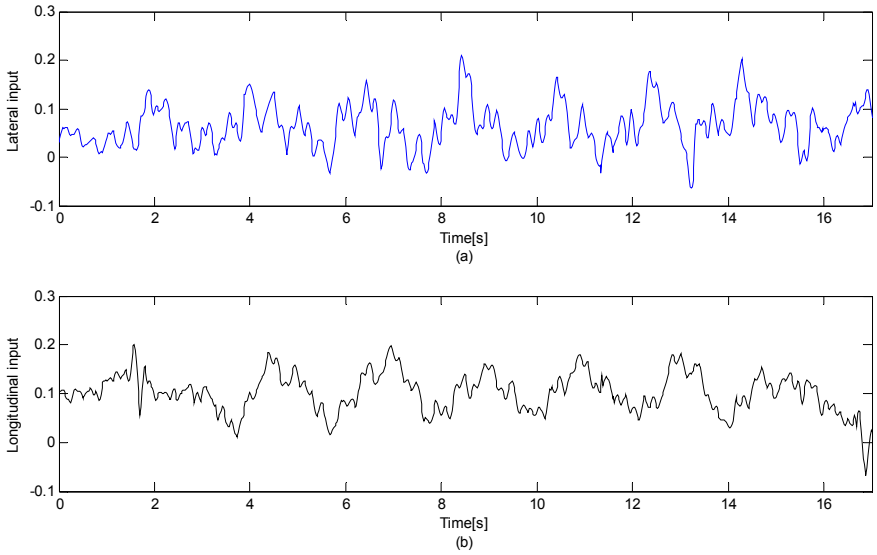


Figure 71: Controller output: (a) lateral command (b) longitudinal command.

7.7 Summary

The effectiveness of the proposed controller is examined by means of intensive experiment carried out on test rig for heading control of helicopter. The non-linearities of the servomotor such as the dead space and stiction are found to be the hidden menace that impedes the controller from achieving a satisfactory steady state regulation performance. Conditional knocker pulse compensation scheme is implemented and is effective in eliminating the oscillatory behaviour due to the stiction phenomenon in the servo motor.

The first motivation for this chapter is that the applicability of DTC-based GPC on helicopter attitude control is investigated through the tracking capability of the control design, the load disturbance rejection performance, and robustness towards unmodelled dynamics. The control objective design parameters of $N = 10, Nu = 2, \lambda = 10, \beta = 0.98$ was found to have reasonably good tracking performance, namely the rise time, overshoot and settling time. The control objective design parameters affect only the tracking performances, while the compromise between the robustness and disturbance rejection performance is shown to be affected solely by the tuning of SP filter.

Secondly, the adaptive SP-based GPC is investigated from the perspective of performance improvement, which is addressed using a robust underlying GPC controller with a robust parameter estimator. The experimental result shows that there are evident improvements in terms of tracking performance and disturbance rejection capability. Specifically, faster settling time in step reference tracking, lower overshoot and faster settling time in disturbance rejection are obtained. More importantly, it also legitimates the use of the underlying SP-based controller that is tuned with lower robustness and allows

additional degree of freedom of tuning between robustness and disturbance rejection for the underlying controller.

The model-based control shows some promising results for the helicopter heading control with very little overshoot and no steady state error.

8. CONCLUSION AND FUTURE WORKS

8.1 Conclusion

In this thesis, modelling of helicopter attitude dynamics is emphasised. Integral-based parameters identification algorithm is used to estimate the unknown intrinsic parameter. The motivation of using simple model in the system identification and for control design results in faster development time and renders the implementation in real-time system that has high sampling rate feasible. Besides, the model mismatch in mathematical modelling is unavoidable as there is no perfect model that fully describes the true system characteristics, especially in high frequency domain. Therefore, the unmodelled high order dynamics is approximated with additional dead time.

The integral method is shown to be effective in identifying both coupled and decoupled models. Thus the method is very flexible and provides the capabilities of adding more complexity if required to capture measured response. The integral method has a further advantage that it can separate the disturbances explicitly from the intrinsic dynamics.

The SP-based GPC being applied to heading control of helicopter in hover mode shows promising results. The control objective to achieve desired tracking performances and disturbance rejection capability with low-order integrative yaw dynamics model plus dead time is obtained. The application of Smith Predictor in the standard GPC is found to achieve satisfactory stability margins in the presence of unmodelled dynamics and time-varying model parameters, namely the input constant, time constant and dead time.

The adaptive SP-based GPC using robust parameters estimator that is implemented on helicopter heading control shows improvement of performance in comparison with the underlying robust controller. The parameters estimator constantly estimates the model parameters that give true nominal model, thus, the performance of the controller is improved even the underlying controller has low robustness. Besides, the range of helicopter operation can be extended to full parameters variation in adaptive control compared with robust controller where the uncertainty range is limited and depends on closed-loop system dynamics.

Another potential robust predictive control approach is investigated where the disturbance is identified in real time and a direct analytical solution computed for the control command allows fast computation and is suitable for fast sampling application. Besides, a simpler and lower computation requirement method is presented for handling actuator constraint instead of solving quadratic programming as is done in MPC.

8.2 Future Works

The following recommendations are intended for the extension of current research. They include two major aspects of improvements. The first part is regarding the experimental setup. The second part is about the system dynamics modelling and the control algorithm.

The experiment on test rig in the current setup is tethered to a computer where the control algorithm computation is done and Bluetooth communication is established between the computer and the onboard controller. The Bluetooth transceiver uses serial

communication UARTs as its data protocol. The normal serial transmission baud rate is about 115.2kbps which is relatively slow for high sampling rate application. The current control sampling rate of 50Hz and data transmission load is just manageable by the serial communication transmission rate. Besides, serial communication does not have data error checking mechanism and there is no data recovery for data transmission loss. Therefore, faster and reliable communication system such as Ethernet should be considered.

For the modelling of translational dynamics of the helicopter, GPS module has to be considered for obtaining the location information in translational plane.

The current modelling is restricted to the attitude dynamics of a scaled helicopter. In future work, attitude parameters will be obtained as functions of dynamic pressure and angle of attack, which is a common approach to modelling translational effects in the literature. Another addition will be including rotor RPM in the attitude parameters. For example, the method presented could be applied to various steady state RPMs to obtain a number of altitude parameter values. These values can be correlated to RPM to bootstrap a more complex model. Hence, the concept is to capture the complexity by interactions of simpler models rather than one very complex model that would be too computationally intense to utilise in real time. Note that when translational dynamics is included, the IMU provides the translational velocities from integrating the accelerometers, so could be easily included in the model if required. A translational model is needed to address guidance of the helicopter to known Global Positioning System (GPS) coordinates and this extension will be investigated in the future as well.

In order to fully validate the present modelling approach to an extension of the current model, flight data from an aggressive execution of flight manoeuvring is required such as benchmark test of helical ascent with coordinated heading change which signifies the non-linearity and coupling of the helicopter dynamics. Then, wider flight envelope model can be obtained and fully validated using the minimal modelling approach. However, the integral method has been shown to be very effective in quite complex models with a large number of parameters (e.g. (Hann, 2005)). Hence further complexity could be added to the modelling as required to capture the measured data while still maintaining very fast and accurate system identification.

Another potential area to be considered for future research would be the synthesis of robust model predictive control that guarantees closed-loop system stability. This can be achieved by solving the optimization problem subject to input constraints and stability constraint. The techniques that enforce stability are divided into two main classes. The first uses the optimization cost function as Lyapunov function, and the second is to shrink the state in some norm. For example, terminal constraint (Kwon and Pearson, 1977) and contraction constraint techniques (Polak and Yang, 1993).

Due to time constraint, the combination control for the roll, pitch and yaw axis channel of helicopter were unable to be implemented while the proof-of-concept of the control algorithm was mostly carried out on the yaw axis channel of the helicopter. However, further control algorithm testing can readily be carried out on the test rig.

References

1) ABBEEL, P., COATES, A. & NG, A. Y. 2010a. Autonomous Helicopter Aerobatics through Apprenticeship Learning. *The International Journal of Robotics Research*.

2) ABBEEL, P., COATES, A. & NG, A. Y. 2010b. Autonomous Helicopter Aerobatics through Apprenticeship Learning. *Int. J. Rob. Res.,* 29**,** 1608-1639.

3) AHMED, B., POTA, H. & GARRATT, M. 2009. Flight control of a rotary wing UAV using backstepping. *International Journal of Robust and Nonlinear Control,* 9999**,** n/a.

4) AHMED, B., POTA, H. R. & GARRATT, M. 2010. Flight control of a rotary wing UAV using backstepping. *International Journal of Robust and Nonlinear Control,* 20**,** 639-658.

5) ANDERSON, B. D. O. & JOHNSTONE, R. M. 1983. Adaptive systems and time varying plants. *International Journal of Control,* 37**,** 367-377.

6) ASTROM, K. Matching criteria for control and identification. Proc. European Control Conference, 1993 Groningen, The Netherlands.

7) ÅSTRÖM, K. J., HAGANDER, P. & STERNBY, J. 1984. Zeros of sampled systems. *Automatica,* 20**,** 31-38.

8) ÅSTRÖM, K. K. J. & HÄGGLUND, T. 2006. *Advanced Pid Control*, ISA-The Instrumentation, Systems, and Automation Society.

9) BAHILL, A. T. 1983. A simple adaptive Smith-predictor for controlling time-delay systems: A tutorial. *Control Systems Magazine, IEEE,* 3**,** 16-22.

10) BAO-CANG, D. 2010. *Modern Predictive Control*, CRC Press.

11) BHANDARI, S., COLGREN, R., LEDERBOGEN, P., & KOWALCHUK, S. six-DoF dynamic modeling and flight testing of a UAV helicopter. Proceedings of the AIAA Modeling and Simulation Technologies Conference and exhibit, 2005 San Francisco, CA. 1-17.

12) BISGAARD, M., LA COUR-HARBO, A. & DIMON BENDTSEN, J. 2010. Adaptive control system for autonomous helicopter slung load operations. *Control Engineering Practice,* 18**,** 800-811.

13) BITMEAD, R. R., GEVERS, M. & WERTZ, V. 1990. *Adaptive optimal control: the thinking man's GPC*, Prentice Hall.

14) BOGDANOV A., C. M., HARVEY G., HUNT J., KIEBURTZ D., VAN DER MERWE R., WAN E. 2003. State-dependent riccati equation control of a small unmanned helicopter. *AIAA Journal of Guidance, Control, and Dynamics*.

15) BRUCE P.D., S. J. E. F., AND KELLETT M.G. 1998. maximum likelihood identification of a rotary-wing RPV simulation model from flight-test data. *Atmospheric Flight Mechanics Conference* Boston, MA.

16) BUSKEY, G., WYETH, G. & ROBERTS, J. Autonomous helicopter hover using an artificial neural network. Robotics and Automation, 2001. Proceedings 2001 ICRA. IEEE International Conference on, 2001 2001. 1635-1640 vol.2.

17) CAMACHO, E. F. & BORDONS, C. 1995. *Model Predictive Control in the Process Industry*, Springer-Verlag.

18) CAMACHO, E. F. & BORDONS, C. 2004. *Model Predictive Control*, Springer London.

19) CASTILLO, C. L., ALVIS, W., CASTILLO-EFFEN, M., MORENO, W. & VALAVANIS, K. Small scale helicopter analysis and controller design for non-aggressive flights. Systems, Man and Cybernetics, 2005 IEEE International Conference on, 10-12 Oct. 2005 2005a. 3305-3312 Vol. 4.

20) CASTILLO, P., LOZANO, R. & DZUL, A. E. 2005b. *Modelling and control of mini-flying machines*, Springer.

21) CHEN, R. T. N. 1979. A Simplified Rotor System Mathematical Model for Piloted Flight Dynamics Simulation. Ames Research Center, NASA.

22) CHOWDHARY, G., LORENZ, S. 2005. Control of a VTOL UAV via Online parameter estimation. *AIAA Journal of Guidance, Control, and Dynamics*.

23) CIVITA, M. L., PAPAGEORGIOU, G., MESSNER, W. C. & KANADE, T. 2006. Design and Flight Testing of an H00 Controller for a Robotic Helicopter. *Journal of Guidance, Control, and Dynamics,* 29**,** 485-494.

24) CLARKE, D. W. 1988. Application of generalized predictive control to industrial processes. *Control Systems Magazine, IEEE*, 8**,** 49-55.

25) CLARKE, D. W. & MOHTADI, C. 1989. Properties of generalized predictive control. *Automatica,* 25**,** 859-875.

26) CLARKE, D. W., MOHTADI, C. & TUFFS, P. S. 1987. Generalized predictive control—Part I. The basic algorithm. *Automatica,* 23**,** 137-148.

27) CLARY, J. P. & FRANKLIN, G. F. A Variable-Dimension Self-Tuning Controller. American Control Conference, 1985, 19-21 June 1985 1985. 973-978.

28) CUTLER, C. & RAMAKER, B. Dynamic matrix control--a computer control algorithm. In Automatic Control Conference, 1980 San Francisco.

29) DICKERSON, L. 2007. UAV on the rise. *Aviation week and space technology, Aerospace Source Book.* New York: McGraw Hill.

30) DOCHERTY, P. D., CHASE, J. G., LOTZ, T. F., HANN, C. E., SHAW, G. M., BERKELEY, J. E., TEMORENGA, L., MANN, J. I. & MCAULEY, K. 2011. Independent cohort cross-validation of the real-time DISTq estimation of insulin sensitivity. *Comput. Methods Prog. Biomed.,* 102**,** 94-104.

31) DOYLE, J. & STEIN, G. 1981. Multivariable feedback design: Concepts for a classical/modern synthesis. *Automatic Control, IEEE Transactions on,* 26**,** 4-16.

32) EGARDT, B. 1979. *Stability of adaptive controllers. Volume 20 of Lecture notes in control and information sciences*, Springer, Berlin.

33) FLETCHER, R. 1981. *Practical Methods of Optimization: Constrained optimization*, J. Wiley.

34) FRAZZOLI, E., DAHLEH, M. A. & FERON, E. Trajectory tracking control design for autonomous helicopters using a backstepping algorithm. American Control Conference, 2000. Proceedings of the 2000, 2000 2000. 4102-4107 vol.6.

35) GADEWADIKAR, J., LEWIS, F., SUBBARAO, K. & CHEN, B. 2008. Structured H command and control-loop design for unmanned helicopters. *AIAA Journal of Guidance, Control, and Dynamics***,** 1093-1102.

36) GADEWADIKAR, J., LEWIS, F., SUBBARAO, K., PENG, K. & CHEN, B. 2009. H-Infinity Static Output-feedback Control for Rotorcraft. *Journal of Intelligent & Robotic Systems,* 54**,** 629-646.

37) GARCIA, C. 2008. Comparison of friction models applied to a control valve. *Control Engineering Practice,* 16**,** 1231-1243.

38) GARCIA, C. E. & MORSHEDI, A. M. 1986. QUADRATIC PROGRAMMING SOLUTION OF DYNAMIC MATRIX CONTROL (QDMC). *Chemical Engineering Communications,* 46**,** 73-87.

39) GARCIA, R. D. & VALAVANIS, K. P. 2009. The Implementation of an Autonomous Helicopter Testbed. *J. Intell. Robotics Syst.,* 54**,** 423-454.

40) GAVRILETS, V., FRAZZOLI, E., METTLER, B., PIEDMONTE, M., & FERON, E. 2001. Aggressive maneuvering of small autonomous helicopters: A human-centered approach. *Robotics Research,* 20(10)**,** 795-807.

41) GAVRILETS, V., MARTINOS, I., METTLER, B. & FERON, E. Control Logic for Automated Aerobatic Flight of a Miniature Helicopter. AIAA Guidance, Navigation, and Control Conference and Exhibit, 2002.

42) GAVRILETS, V., METTLER, B., & FERON, E. 2004. Human-inspired control logic for automated maneuvering of miniature helicopter. *AIAA Journal of Guidance, Control, and Dynamics,* 27(5)**,** 752-759.

43) GERRY, J. J., & RUEL, M. 2001. How to measure and combat valve stiction online. Houston, USA:ISA.

44) GIRI, F., M'SAAD, M., DION, J. M. & DUGARD, L. 1991. On the robustness of discrete-time indirect adaptive (linear) controllers. *Automatica,* 27**,** 153-159.

45) GOODWIN, G. C. & SIN, K. S. 1984. *Adaptive filtering prediction and control*, Prentice-Hall.

46) GUTMAN, P.-O. 1994. On-line parameter interval estimation using recursive least squares. *International Journal of Adaptive Control and Signal Processing,* 8**,** 61-72.

47) HÄGGLUND, T. 2002. A friction compensator for pneumatic control valves. *Journal of Process Control,* 12**,** 897-904.

48) HANBO, Q., GUANQING, C., HONGXING, C. & YIPING, Y. A Grey-Modeling Research on a Small-Scale Autonomous Helicopter. Information Engineering and Computer Science, 2009. ICIECS 2009. International Conference on, 19-20 Dec. 2009 2009. 1-4.

49) HANN, C. E., CHASE, J.G., LIN, J., LOTZ, T., DORAN, C.V., SHAW, G.M 2005. Integral-based parameter identification for long-term dynamic verification of a glucose-insulin system model. *Computer Methods and Programs in Biomedicine,* 77(3)**,** 259-270.

50) HANN, C. E., CHASE, J.G., YPMA, M.F., ELFRING, J., NOR, N.M.H., LAWRENCE, P., SHAW, G.M. 2008. The Impact of Parameter Identification Methods on Drug Therapy Control in an Intensive Care Unit. *The Open Medical Informatics Journal,* 2**,** 92-104.

51) HANN, C. E., M SNOWDON, A RAO, O WINN, N WONGVANICH, X CHEN 2012. Minimal modelling approach to describe turbulent rocket roll dynamics in a vertical wind tunnel. *Journal of Aerospace Engineering,* 226**,** 1042-1060.

52) HYUNCHUL SHIM, D., HYOUN JIN, K. & SASTRY, S. Control system design for rotorcraft-based unmanned aerial vehicles using time-domain system identification. Control Applications, 2000. Proceedings of the 2000 IEEE International Conference on, 2000 2000. 808-813.

53) JAKOBY, W. & PANDIT, M. 1987. A prediction-error-method for recursive identification of nonlinear systems. *Automatica,* 23**,** 491-496.

54) JOHNSON, E., & KANNAN, S. 2005. Adaptive Trajectory Control for Autonomous Helicopters. *AIAA Journal of Guidance, Control, and Dynamics,* 28(3)**,** 524-538.

55) KALLAPUR, A. G. & ANAVATTI, S. G. UAV Linear and Nonlinear Estimation Using Extended Kalman Filter. Computational Intelligence for Modelling, Control and Automation, 2006 and International Conference on Intelligent Agents, Web Technologies and Internet Commerce, International Conference on, Nov. 28 2006-Dec. 1 2006 2006. 250-250.

56) KANADE, M. L. C. A. A. W. C. M. A. T. Modeling of Small-Scale Helicopters with Integrated First-Principles and System-Identification Techniques. Proceedings of the 58th Forum of the American Helicopter Society, 2002a Montreal, Canada. 2505-2516.

57) KANADE, M. L. C. A. A. W. C. M. A. T. Modeling of Small-Scale Helicopters with Integrated First-Principles and System-Identification Techniques. 2002b Montreal, Canada. 2505-2516.

58) KENDOUL, F. 2012. Survey of advances in guidance, navigation, and control of unmanned rotorcraft systems. *Journal of Field Robotics,* 29**,** 315-378.

59) KENNÉ, G., AHMED-ALI, T., LAMNABHI-LAGARRIGUE, F. & NKWAWO, H. 2006. Nonlinear systems parameters estimation using radial basis function network. *Control Engineering Practice,* 14**,** 819-832.

60) KIM, H. J., SHIM, D. H. & SASTRY, S. Nonlinear model predictive tracking control for rotorcraft-based unmanned aerial vehicles. American Control Conference, 2002. Proceedings of the 2002, 2002 2002. 3576-3581 vol.5.

61) KIM, J. & SHIM, D. 2003. A flight control system for aerial robots: algorithms and experiments. *Control Engineering Practice,* 11**,** 1389-1400.

62) KIM, S. K. & TILBURY, D. M. 2004. Mathematical Modeling and Experimental Identification of an Unmanned Helicopter Robot with Flybar Dynamics. *Journal of robotic systems,* 21**,** 95-116.

63) KOO, T. J. & SASTRY, S. Output tracking control design of a helicopter model based on approximate linearization. Decision and Control, 1998. Proceedings of the 37th IEEE Conference on, 16-18 Dec 1998 1998. 3635-3640 vol.4.

64) KWON, W. H. & PEARSON, A. E. 1977. A modified quadratic cost problem and feedback stabilization of a linear system. *Automatic Control, IEEE Transactions on,* 22**,** 838-842.

65) LA CIVITA, M., PAPAGEORGIOU, G., MESSNER, W. C., & KANADE, T. 2006. Design and flight testing of an H controller for a robotic helicopter. *AIAA Journal of Guidance, Control, and Dynamics*, 485-494.

66) LANDAU, I. D., LOZANO, R., M'SAAD, M. & KARIMI, A. 2011. Adaptive Control. 2 ed. Dordrecht: Springer.

67) LANDAU, Y. D. 1979. *Adaptive Control: The Model Reference Approach*, Marcel Dekker, Inc.

68) LEONTARITIS, I. J. & BILLINGS, S. A. 1985. Input-output parametric models for non-linear systems Part I: deterministic non-linear systems. *International Journal of Control,* 41, 303-328.

69) LJUNG, L. 1987. *System identification: theory for the user*, Prentice-Hall.

70) LJUNG, L. & SÖDERSTRÖM, T. 1985. *Theory and practice of recursive identification*, MIT Press.

71) LONDON, U. V. 2009. The unmanned aerial vehicle (UAV) market 2009-2019

72) LUENBERGER, D. G. 1984. *Linear and nonlinear programming*, Addison-Wesley.

73) LYASHEVSKIY, S. & YAOBIN, C. Nonlinear identification of aircraft. Control Applications, 1996., Proceedings of the 1996 IEEE International Conference on, 15-18 Sep 1996 1996. 327-331.

74) M.RUEL 2000. Stiction:the hidden menace. *Control Magazine.*

75) MACIEJOWSKI, J. M. 2002. *Predictive Control: With Constraints*, Prentice Hall.

76) MAHONY, R., HAMEL, T. & DZUL, A. Hover control via Lyapunov control for an autonomous model helicopter. Decision and Control, 1999. Proceedings of the 38th IEEE Conference on, 1999 1999. 3490-3495 vol.4.

77) MARTINI, A., LÉONARD, F. & ABBA, G. 2009. Dynamic Modelling and Stability Analysis of Model-Scale Helicopters Under Wind Gust. *Journal of Intelligent & Robotic Systems,* 54, 647-686.

78) METTLER, B. 2003. *Identification modeling and characteristics of miniature rotorcraft*, Kluwer Academic Publishers.

79) METTLER, B., KANADE, T. & TISCHLER, M. B. 2000. System identification modeling of a model-scale helicopter. Pittsburgh, PA: Robotics Institute, Carnegie Mellon University.

80) MONTGOMERY, J. F. & BEKEY, G. A. Learning helicopter control through teaching by showing Proceedings of the 37th IEEE Conference on Decision and Control, 16-18 Dec 1998 1998. 3647-3652 vol.4.

81) MORARI, M. & ZAFIRIOU, E. 1989. *Robust Process Control*, Prentice Hall.

82) MORENO, W. A., RAPTIS, I. A. & VALAVANIS, K. P. 2009. System Identification and Discrete Nonlinear Control of Miniature Helicopters Using Backstepping. *Journal of intelligent & robotic systems,* 55**,** 223-243.

83) MORRIS, J. C., VAN NIEUWSTADT, M. & BENDOTTI, P. Identification and control of a model helicopter in hover. American Control Conference, 1994, 29 June-1 July 1994 1994. 1238-1242 vol.2.

84) MUTAMBARA, A. G. O. & AL-HAIK, M. S. Y. EKF based parameter estimation for a heat exchanger. American Control Conference, 1999. Proceedings of the 1999, 1999 1999. 3918-3922 vol.6.

85) NORMEY-RICO, J. E. 2007. *Control of Dead-time Processes*, Springer.

86) NORMEY-RICO, J. E., GÓMEZ-ORTEGA, J. & CAMACHO, E. F. 1999. A Smith-predictor-based generalised predictive controller for mobile robot path-tracking. *Control Engineering Practice,* 7**,** 729-740.

87) OLFATI-SABER, R. 2001. *Nonlinear control of underactuated mechanical systems with application to robotics and aerospace vehicles.* PhD thesis, Massachusetts Institute of Technology.

88) OLLERO, A. & MERINO, L. 2004. Control and perception techniques for aerial robotics. *Annual Reviews in Control,* 28**,** 167-178.

89) PADFIELD, G. D. 2007. *Helicopter flight dynamics: the theory and application of flying qualities and simulation modelling*, Blackwell Publishing.

90) PENG, K., CAI, G., CHEN, B. M., DONG, M., LUM, K. Y. & LEE, T. H. 2009. Design and implementation of an autonomous flight control law for a UAV helicopter. *Automatica,* 45**,** 2333-2338.

91) PETERKA, V. 1984. Paper: Predictor-based self-tuning control. *Automatica,* 20**,** 39-50.

92) PETERSON, B. & NARENDRA, K. S. 1982. Bounded error adaptive control. *Automatic Control, IEEE Transactions on,* 27**,** 1161-1168.

93) POLAK, E. & YANG, T. H. 1993. Moving horizon control of linear systems with input saturation and plant uncertainty Part 2.Disturbance rejection and tracking. *International Journal of Control,* 58**,** 639-663.

94) PRALY, L. Robustness of indirect adaptive control based on pole placement design. Proc. 1st IFAC Workshop on Adaptive Systems in Control and Signal Processing,, 1983 San Francisco, CA.

95) PROUTY, R. W. 1990. *Helicopter performance, stability, and control*, R.E. Krieger Pub. Co.

96) PROUTY, R. W. 1995. *Helicopter performance, stability, and control*, Krieger Pub.

97) RAOL, J. R., GIRIJA, G., SINGH, J. & ENGINEERS, I. O. E. 2004. *Modelling and parameter estimation of dynamic systems*, Institution of Electrical Engineers.

98) RAPTIS, I., VALAVANIS, K. & MORENO, W. 2009a. System Identification and Discrete Nonlinear Control of Miniature Helicopters Using Backstepping. *Journal of Intelligent and Robotic Systems,* 55**,** 223-243.

99) RAPTIS, I. A., VALAVANIS, K. P. & MORENO, W. A. 2009b. System Identification and Discrete Nonlinear Control of Miniature Helicopters Using Backstepping. *J. Intell. Robotics Syst.,* 55**,** 223-243.

100) RAWLINGS, J. B. & MAYNE, D. Q. 2009. *Model Predictive Control: Theory and Design*, Nob Hill Publishing.

101) RICHARD, J.-P. 2003. Time-delay systems: an overview of some recent advances and open problems. *Automatica,* 39**,** 1667-1694.

102) ROSSITER, J. A. 2013. *Model-Based Predictive Control: A Practical Approach*, Taylor & Francis.

103) SALMAN, S. A., PUTTIGE, V. R. & ANAVATTI, S. G. Real-time validation and comparison of fuzzy identification and state-space identification for a UAV platform. Computer Aided Control System Design, 2006 IEEE International Conference on Control Applications, 2006 IEEE International Symposium on Intelligent Control, 2006 IEEE, 4-6 Oct. 2006 2006. 2138-2143.

104) SHIN, J., NONAMI, K., FUJIWARA, D. & HAZAWA, K. 2005. Model-based optimal attitude and positioning control of small-scale unmanned helicopter. *Robotica,* 23**,** 51-63.

105) SILVA, G. J., DATTA, A. & BHATTACHARYYA, S. P. Controller design via Pade approximation can lead to instability. Decision and Control, 2001. Proceedings of the 40th IEEE Conference on, 2001 2001. 4733-4737 vol.5.

106) SKOGESTAD, S. & POSTLETHWAITE, I. 2005. *Multivariable feedback control: analysis and design*, John Wiley.

107) SMERLAS, A., I, P., WALKER, D., STRANGE, M., HOWITT, J., NORTON, R., GUBBELS, A., & BAILLIE, S. . Design and Flight Testing of an H Sub(infinity) Controller for the NRC Bell 205 Experimental Fly-by-Wire Helicopter. Proceedings of the AIAA Guidance, Navigation and Control Conference, 1998 Boston MA.

108) SUGENO, M., HOWARD, W., ISAO, H., & SATORU, K. Intelligent control of an unmanned helicopter based on fuzzy logic. Proceedings of the 51st American Helicopter Society (AHS) Annual Forum, 1995 Fort Worth, TX. 791-803.

109) TAKAHASHI, M., SCHULEIN, G., & WHALLEY, M. Flight control law design and development for an autonomous rotorcraft. Proceedings of the 64th Annual Forum of the American Helicopter Society, 2008 Montreal, Quebec. 1652-1671.

110) TANNER, O. 2003. *Modeling, Identification and Control of Autonomous Helicopters.* PhD Dissertation, Swiss Federal Institute of Technology, Zurich, Switzerland.

111) TISCHLER, B. M. A. T. K. A. M. B. 2000. System identification modeling of a model-scale helicopter. Journal of the American Helicopter Society.

112) TISCHLER, M. B. & REMPLE, R. K. 2006a. *Aircraft And Rotorcraft System Identification: Engineering Methods With Flight-test Examples*, American Institute of Aeronautics and Astronautics.

113) TISCHLER, M. B. & REMPLE, R. K. 2006b. *Aircraft and Rotorcraft System Identification: Engineering Methods With Flight-Test Examples*, AIAA.

114) UNDERWOOD, S. J. & HUSAIN, I. 2010. Online Parameter Estimation and Adaptive Control of Permanent-Magnet Synchronous Machines. *IEEE transactions on industrial electronics (1982),* 57**,** 2435-2443.

115) VAHIDI, A., STEFANOPOULOU, A. & PENG, H. 2005. Recursive least squares with forgetting for online estimation of vehicle mass and road grade: theory and experiments. *Vehicle System Dynamics,* 43**,** 31-55.

116) VALAVANIS, K. P. 2008. *Advances in Unmanned Aerial Vehicles: State of the Art and the Road to Autonomy*, Springer.

117) WAN, E. A. & BOGDANOV, A. A. Model predictive neural control with applications to a 6 DOF helicopter model. American Control Conference, 2001. Proceedings of the 2001, 2001 2001. 488-493 vol.1.

118) WANG, L. 2009. *Model Predictive Control System Design and Implementation Using MATLAB®*, Springer.

119) WEILENMANN, M. F. & GEERING, H. P. 1994. Test bench for rotorcraft hover control. *Journal of Guidance, Control, and Dynamics,* 17**,** 729-736.

120) WONG, X. W., CHASE, J. G., SHAW, G. M., HANN, C. E., LOTZ, T., LIN, J., SINGH-LEVETT, I., HOLLINGSWORTH, L. J., WONG, O. S. W. & ANDREASSEN, S. 2006. Model predictive glycaemic regulation in critical illness using insulin and nutrition input: A pilot study. *Medical Engineering & Physics,* 28**,** 665-681.

121) XIANG IVAN, L. Z. & LAKSHMINARAYANAN, S. 2009. A New Unified Approach to Valve Stiction Quantification and Compensation. *Industrial & Engineering Chemistry Research,* 48**,** 3474-3483.

122) YOON, T.-W. & CLARKE, D. W. 1995. Observer design in receding-horizon predictive control. *International Journal of Control,* 61**,** 171-191.